



TESIS KI142502

**DENSE VISUAL WORD SPATIAL ARRANGEMENT
DAN PENERAPANNYA BERSAMA FITUR WARNA
DAN TEKSTUR PADA PENGENALAN OBJEK
SECARA OTOMATIS**

Gama Wisnu Fajarianto
5112201006

DOSEN PEMBIMBING
Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015



THESIS KI142502

**DENSE VISUAL WORD SPATIAL ARRANGEMENT
AND ITS APPLICATION WITH THE COLOR AND
TEXTURE FEATURES ON AUTOMATIC OBJECT
RECOGNITION**

Gama Wisnu Fajarianto
5112201006

SUPERVISOR
Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

MAGISTER PROGRAMME
INTELLIGENCE COMPUTATIONAL AND VISUALIZATION
INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2015

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya


oleh:
Gama Wisnu Fajarianto
Nrp. 5112201006

Dengan Judul :
Dense Visual Word Spatial Arrangement Dan Penerapannya
Bersama Fitur Warna Dan Tekstur Pada Pengenalan Objek Secara
Otomatis


Tanggal Ujian : 15 Januari 2015
Periode Wisuda : 2015 Gasal

Disetujui oleh:


1. Prof.Ir.Handayani Tjandrasa ,M.Sc, Ph.D
NIP. 194908231976032001


.....
(Pembimbing I)


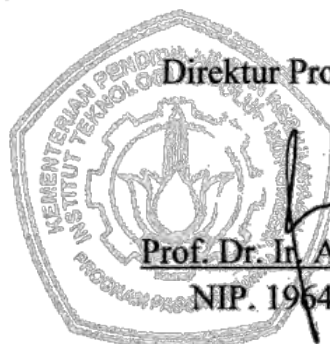
2. Dr. Darlis Heru Murti, S.Kom, M.Kom.
NIP. 197712172003121001


.....
(Penguji 1)

3. Anny Yuniarti, S.Kom.,M.Comp.Sc
NIP. 198106202005011003


.....
(Penguji 2)

4. Wijayanti Nurul Khotimah, S.Kom., M.Sc.
NIP. 198603122012122004


.....
(Penguji 3)

Direktur Program Pascasarjana,


Prof. Dr. In. Adi Soeprijanto, MT
NIP. 19640405 199002 1 001

**DENSE VISUAL WORD SPATIAL ARRANGEMENT
DAN PENERAPANNYA BERSAMA FITUR WARNA
DAN TEKSTUR PADA PENGENALAN OBJEK
SECARA OTOMATIS**

Nama mahasiswa : Gama Wisnu Fajariato.
NRP : 5112201006
Pembimbing : Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D

ABSTRAK

Bag of visual word (BoVW) merupakan metode yang menjelaskan isi dari gambar. Metode ini hanya menghitung banyaknya word dan tidak memberikan informasi spasial. Terdapat metode Visual word spatial arrangement (WSA) dimana metode ini memberikan informasi spasial tentang word tertentu pada gambar dengan menggunakan interest point sebagai detektor.

WSA kurang dapat memberikan informasi yang penting pada gambar dikarenakan interest point yang dihasilkan oleh detektor dapat memberikan titik-titik yang berpotensi tidak merupakan representasi yang penting dari gambar tersebut. Pada tesis ini diusulkan metode dense visual word spatial arrangement (DVSA) yang merupakan modifikasi metode dari WSA. Metode ini tidak menggunakan detektor interest point untuk menghitung deskriptor lokal melainkan dengan menghitung deskriptor lokal pada bagian komponen piksel-piksel yang saling berdekatan.

Hasil pengujian pada 4485 gambar dengan 15 jenis kelas menggunakan 10-fold cross validation untuk 2 word metode yang diusulkan memberikan peningkatan performa sebesar 12.68 % dari akurasi BoVW sedangkan akurasi WSA lebih baik 15.62 % dari BoVW. Untuk 4 word metode yang diusulkan memberikan peningkatan performa akurasi sebesar 30.99 % dari akurasi BoVW dan peningkatan performa 18.16 % dari WSA. Sedangkan untuk 6 word metode yang diusulkan memberikan peningkatan performa sebesar 29.98 % dari akurasi BoVW dan peningkatan performa 18.75 % dari WSA. Peningkatan performa akurasi sebesar 36.2 % didapatkan oleh metode yang diusulkan dengan 6 word terhadap BoVW

dengan 2 word. Peningkatan performa sampai 18.75 % yang dihasilkan DVSA dibandingkan WSA dan peningkatan performa sampai 30.99 % dibandingkan BoVW dengan jumlah word yang sama menunjukkan metode yang diusulkan kompetitif untuk mengenali jenis gambar.

Kata kunci: deskriptor lokal, visual word, klasifikasi, ekstraksi fitur

DENSE VISUAL WORD SPATIAL ARRANGEMENT AND ITS APPLICATION WITH THE COLOR AND TEXTURE FEATURES ON AUTOMATIC OBJECT RECOGNITION

ABSTRACT

Bag of visual word (BoVW) is a method that describes the contents of an image. This method simply counts the number of words, but it doesn't provide spatial information. Besides there is a method that provides spatial information about particular words in the image by using an interest point as a detector. The method is Visual word spatial arrangement (WSA).

WSA can provide less important information on the image generated due to the interest point doesn't represent the main aspects of the image. In this thesis, Dense visual word spatial arrangement (DVSA) method which is proposed is a modification of the WSA method. The proposed method doesn't use an interest point detector to compute local descriptor but it uses a local descriptor that computes at the component pixels adjacent to each other.

The test result on 4485 images with 15 types of classes is computed using 10 fold cross validation for 2 words of the proposed method that provides an improved performance by 12.68% of accuracy BoVW, while WSA has better accuracy by 15.62% from BoVW. For 4 words, the proposed method provides an improved performance by 30.99% from the accuracy of BoVW, and an improved performance by 18.16% from WSA. While for 6 words, the proposed method provides an improved performance by 29.98% from the accuracy of BoVW, and an improved performance by 18.75% from WSA. The improved performance of the accuracy by 36.20% is obtained by the proposed method with 6 words than BoVW with 2 words. From the result can be concluded that the proposed method or DVSA method is more competitive to recognize images.

Keywords: local descriptor, visual word, classification, fitur extraction

KATA PENGANTAR

Alhamdulillahirobila'alamiin, puji syukur kehadiran Allah SWT atas segala nikmat dan karunianya. Sehingga tesis dengan judul **“DENSE VISUAL WORD SPATIAL ARRANGEMENT DAN PENERAPANNYA BERSAMA FITUR WARNA DAN TEKSTUR PADA PENGENALAN OBJEK SECARA OTOMATIS”** dapat terselesaikan dengan baik.

Tesis ini diselesaikan guna memenuhi persyaratan untuk memperoleh gelar Magister Komputer (M. Kom.) dalam bidang keahlian Komputasi Cerdas dan Visualisasi pada program studi Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Tak lupa penulis ucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang berperan dalam terselesaikannya tesis ini diantaranya:

1. Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D, selaku dosen pembimbing, yang dengan sabar memberikan bimbingan dan arahan serta waktu guna terselesaikannya tesis ini.
2. Bapak Dr. Darlis Heru Murti, S.Kom., M.Kom., Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom., Ibu Anny Yuniarti, S.Kom., M.Comp.Sc., dan Ibu Wijayanti Nurul Khotimah, S.Kom., M.Sc., selaku penguji selama sidang proposal dan tesis atas masukan dan saran serta bimbingannya dengan sabar guna perbaikan.
3. Prof. Dr. Ir. Adi Supriyanto, M.T., selaku Direktur Program Pascasarjana beserta dosen dan karyawan di lingkungan Pascasarjana.
4. Bapak dan Ibu Dosen yang dengan sabar membimbing, mengarah dan mengajarkan ilmunya beserta Karyawan dan Staf Teknik Informatika ITS, selama penulis menempuh pendidikan di Teknik Informatika ITS.
5. Kedua orang tua penulis, Bapak Sri Handono dan Ibu Hesti Udjianti yang dengan sabar memberikan dukungan do'a dan moril secara terus menerus tanpa penulis memintanya.
6. Kakak dan Adik penulis Hertiana Betaningtyas dan Deltaningtyas Tri Cahyaningrum yang memberikan dukungan do'a dan motivasi.

7. Saudara penulis yang tidak bisa penulis sebut satu persatu yang memberikan motivasi sehingga terselesaikan tesis ini.
8. Teman penulis di kampung halaman Trias, Mas Geri , Ubub, Andik, Muzammil, Elga, Bagus, Fahmi dan semuanya yang secara tidak langsung memberikan motivasi kepada penulis.
9. Teman senasib dan seperjuangan beasiswa Fresh Graduate, Mas Indra, Lutfi, Tesa, dan Irsyad yang semasa menempuh pendidikan membantu penulis baik dikala susah atau senang.
10. Bapak/Ibu/Mbak/Mas Abror, Daniel, Nur Hayatin, Dany, Fadil, Aminuddin, Aditya, Parma, Sofyan, Geges, Yusuf, Noor, Nanang, Evy, Mustika, Yuita, Lukman, Siti, Alif dan semua rekan-rekan satu jurusan S-2 Teknik Informatika angkatan 2012.
11. Teman-teman Lab S-1 Aminuddin, Ridwan, Hayam, Alrezza, Aldy, Ade, Hasfi, Sindu, Angga, Ampuh, Puguh, Rimbi, Agus, Ghazi, Yusuf, Irna, Febi, Nabilla, Ira dan semuanya yang membantu penulis selama penulis menempuh pendidikan di Teknik Informatika ITS.
12. Teman-teman penulis Hisyam, Jamal, Yusuf, Rifai, Afif, Radik, Hima, Rifky, Yoga, Septian, Kadiq, Bagus, Hadhori, Fanji dan semuanya yang membantu memberikan ilmu dan motivasi kepada penulis.
13. Keluarga HMTC, JMMI ITS, KMI yang baik secara langsung maupun tidak langsung memberikan bantuan kepada penulis.
14. Rektor dan semua civitas akademika Institut Teknologi Sepuluh Nopember, jurusan Teknik Informatika pada khususnya.

Dengan segala kekurangan dan keterbatasan baik dalam hal pengalaman, pengetahuan dan pustaka yang penulis miliki, penulis menyadari tesis ini masih banyak kekurangan, oleh karena itu penulis mengharap kritik yang membangun guna perbaikan dalam penyusunan laporan selanjutnya.

Akhir kata, penulis berharap tesis ini bisa bermanfaat bagi kita semua terutama kepada pembaca dan terhadap pengembangan ilmu pengetahuan di bidang Komputasi Cerdas dan Visualisasi

Surabaya, Januari 2015

DAFTAR ISI

ABSTRAK	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	4
1.3 Tujuan dan Manfaat Penelitian.....	5
1.4 Batasan Masalah.....	5
1.5 Kontribusi	5
BAB 2 KAJIAN PUSTAKA	7
2.1 Bag of Visual Word, dan Visual <i>Word</i> Spatial Arrangement	7
2.2 Bag of Visual Word.....	18
2.3 Visual Word Spatial Arrangement	20
2.3.1 Interest Point	20
2.3.2 Detektor Scale-Invariant Feature Transform	20
2.3.3 Deskriptor Scale-Invariant Feature Transform	21
2.4 Dense SIFT.....	22
2.5 Evaluasi	22
2.5.1 Presisi	23
2.5.2 Recall atau True Positif Rate (TPRate)	23
2.5.3 F-Measure (F_1)	23
BAB 3 METODE PENELITIAN	25

3.1	Rancangan Penelitian.....	25
3.1.1.	Pengerjaan pada dataset citra umum	25
3.1.2.	Pengerjaan pada dataset citra bunga.....	26
3.2	Skenario Uji Coba.....	26
3.3	Langkah-langkah Penelitian	27
3.3.1.	Pemilihan dataset.....	28
3.3.2.	Pengelompokan secara manual jenis dataset	32
3.3.3.	Visual Dictionary.....	32
3.3.4.	Dense Visual Word Spatial Arrangement	32
3.3.5.	Segmentasi Citra.....	33
3.3.6.	Ekstraksi Fitur	38
3.3.7.	Evaluasi	38
BAB 4	IMPLEMENTASI DAN PEMBAHASAN	44
4.1	Implementasi.....	44
4.1.1.	Membuat Dictionary.....	44
4.1.2.	Mendapatkan Word dan Posisinya	46
4.1.3.	Bag of Visual Word.....	48
4.1.4.	Visual Word Spatial Arrangement	49
4.1.5.	Dense Visual Word Spatial Arrangement	50
4.2	Uji Coba dan Hasil.....	52
4.2.1.	Analisa Uji Coba dan Hasil Secara Keseluruhan	52
4.2.2.	Hasil Metode yang diusulkan terhadap BoVW dan WSA	56
4.2.3.	Hasil Peningkatan Performa pada DVSA, BoVW dan WSA.....	56
4.2.4.	Hasil Performa Penerapan DVSA, BoVW dan DVSA bersama Fitur Warna dan Tekstur pada Pengenalan Objek Bunga	57
BAB 5	KESIMPULAN DAN SARAN	62

5.1	Kesimpulan.....	62
5.2	Saran.....	62
DAFTAR PUSTAKA		64
BIODATA PENULIS		65

DAFTAR TABEL

Tabel 2.1 Confusion matrix.....	24
Tabel 3.1 Label dataset 15-scenes yang digunakan	39
Tabel3.2 Label dataset citra 30 jenis bunga digunakan beserta nama jenisnya	31
Tabel 4.1Hasil Evaluasi Metode BoVW, WSA dan DVSA	55
Tabel 4.2 Hasil Peningkatan Metode untuk Setiap <i>Word</i>	56
Tabel 4.3 Hasil Evaluasi Penggabungan Fitur <i>word</i> dengan Fitur Warna dan Tekstur.....	59

DAFTAR GAMBAR

Gambar 1.1. Bag of visual words tidak memberikan informasi geometris	2
Gambar 1.2. Dengan tidak adanya informasi geometris pada bag of visual words jenis gambar yang berbeda dapat memiliki bag of visual words yang mirip	3
Gambar 1.3 Interest point pada MITopencountry dan MITcoast serta MITtallbuilding dan industrial pada visual <i>word</i> spatial arrangement)	4
Gambar 2.1. Hutan yang dilihat dari atas	7
Gambar 2.2. Hutan yang memiliki sungai	8
Gambar 2.3. Hutan pada musim salju	9
Gambar 2.4. Hutan dengan air terjun	9
Gambar 2.5. Hutan dengan hewan di dalamnya	10
Gambar 2.6. Gambar gunung dengan pendaki gunung	10
Gambar 2.7. Gunung dengan asapnya	11
Gambar 2.8. Gunung dengan langitnya terdapat objek terang	11
Gambar 2.9. Gunung dengan sungainya	12
Gambar 2.10. Gunung dengan sungai dan terdapat bangunan disekitarnya	12
Gambar 2.11. Gunung dengan perumahan disekitarnya dan cahaya bulan	13
Gambar 2.12. Gunung dengan pendaki yang menuruni gunung	13
Gambar 2.13. Toko dengan penjualan topi	14
Gambar 2.14. Toko dengan penjualan sepatu	14
Gambar 2.15. Toko dengan buku-bukunya	15
Gambar 2.16. Toko dengan orang di depan penjualan	15
Gambar 2.17. Toko dengan penjualan yang tertata di rak	16
Gambar 2.18. Perindustrian dengan cerobong asapnya	16
Gambar 2.19. Contoh gambar jenis ikan lele	17
Gambar 2.20. Contoh gambar jenis ikan Mujair	17
Gambar 2.21. Gambar diagram ekstraksi fitur dalam mengenali jenis gambar	18
Gambar 2.22. Diagram metode <i>word</i> dalam mengenali jenis gambar	19
Gambar 2.23. Diagram alir dari bag of visual <i>word</i>	19
Gambar 2.24. Matrix untuk proses coding dan pooling	20

Gambar 2.25. 3D Spatial Histogram SIFT	21
Gambar 2.26. Orientasi pada SIFT dan jumlahnya untuk setiap subblok	22
Gambar 2.27. Ilustrasi detektor dense SIFT pada potongan citra	22
Gambar 3.1 Diagram alir tahapan metodologi	26
Gambar 3.2 Contoh citra dari dataset 15-scenes	28
Gambar 3.3 Dataset citra 30 jenis bunga yang digunakan beserta label	30
Gambar 3.4 Histogram jumlah dan jenis citra bunga.	31
Gambar 3.5 Diagram alir tahapan membuat visual dictionary	33
Gambar 3.6 Diagram alir tahapan metode yang diajukan	34
Gambar 3.7 Diagram alir tahapan untuk mendapatkan jendela objek.....	36
Gambar 3.8 Diagram tahapan ekstraksi fitur	38
Gambar 3.9 Diagram tahapan evaluasi pertama	39
Gambar 3.10 Diagram tahapan evaluasi pada citra umum	40
Gambar 3.11 Diagram tahapan evaluasi secara detail	41
Gambar 4.1 Hasil confusion matrix tanpa tambahan fitur word	57
Gambar 4.2 Hasil confusion matrix dengan tambahan fitur word dari WSA	58
Gambar 4.3 Hasil confusion matrix dengan tambahan fitur word dari DVSA	59
Gambar 4.4 Hasil confusion matrix dengan tambahan fitur word dari BoVW	60

BAB 1

PENDAHULUAN

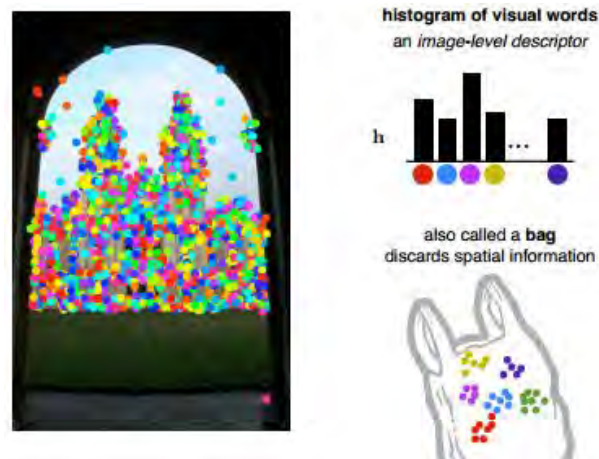
1.1 Latar Belakang

Salah satu pendekatan populer dalam menjelaskan isi dari gambar adalah dengan pendekatan *bag of visual word* (Dimitrovski et al., 2014; Koniusz et al., 2013; López-Sastre et al., 2013; Sánchez et al., 2012; Zhang et al., 2013). Selain populer pendekatan ini juga efektif dalam menjelaskan isi dari gambar (Penatti et al., 2014; Bolovinou et al., 2013; Sánchez et al., 2012). Pendekatan *bag of visual word* dapat dijelaskan secara umum dengan 3 langkah yaitu pertama *local image descriptor* diekstraksi dari gambar, kemudian *visual dictionary* didapat dari sekumpulan fitur vektor *local image descriptor* yang bisa diperoleh dengan menggunakan *clustering k-means*. Langkah kedua adalah *fitur encoding* yaitu mengaktifkan *visual word* dengan memetakan *fitur descriptor* ke *visual dictionary*, dan langkah ketiga yaitu *pooling* dimana adalah langkah yang menjadikan hasil dari *encoding fitur descriptor* dengan *visual dictionary* menjadi satu fitur vektor (Koniusz et al., 2013; Avila et al., 2013).

Aplikasi yang menggunakan *bag of visual words* dapat ditemui seperti pada aplikasi yang dapat membedakan tulisan tangan dengan tulisan teks mesin *print* pada suatu dokumen (Zagoris et al., 2014). Kemudian aplikasi *scene categorization*, contohnya *keyword suggestion* yaitu menawarkan beberapa *label* yang berhubungan dengan isi gambar. Dan aplikasi lain adalah aplikasi *retrieval* yaitu *memfilter* gambar pada internet berdasarkan pada *keyword* (Li and Yap, 2013). Aplikasi berikutnya dapat ditemui pada bidang *biomedical engineering* yaitu otomatisasi analisa dari *time series* biomedis *electroencephalogram (EEG)* dan *electrocardiographic (ECG) signal* dimana *bag of visual words* digunakan untuk merepresentasikan *biomedical time series* (J. Wang et al., 2013).

Representasi gambar dengan *bag of visual words* tidak memberikan informasi geometris dari gambar (Zhang et al., 2013; Bolovinou et al., 2013; Penatti

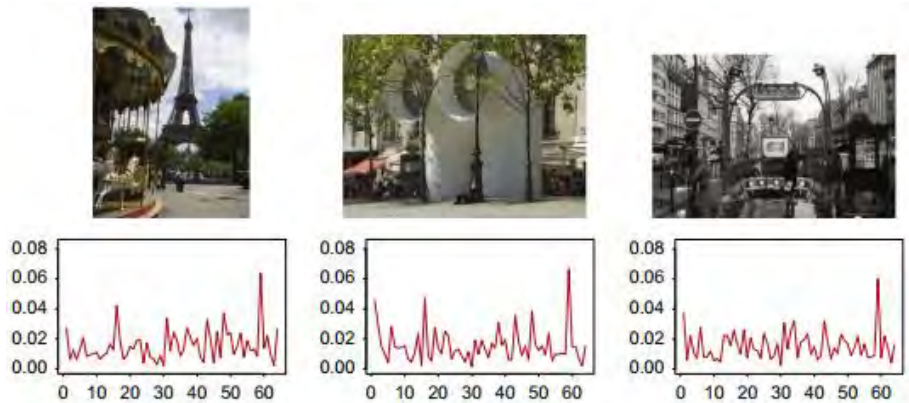
et al., 2014; Sánchez et al., 2012). Sebagai representasi *bag of visual words* ditunjukkan pada Gambar 1.1 dibawah. Bulatan warna menunjukkan representasi dari *word*. Sedangkan histogram yang merupakan jumlahan dari banyaknya *word* untuk setiap warna ditunjukkan pada bagian *histogram of visual words* pada Gambar 1.1. Pada Gambar 1.1 digambarkan juga mengenai karakter dari *bag of visual words* seperti *word-word* yang berada dalam tas yang merepresentasikan bahwa *bag of visualword* menghilangkan informasi *spatial* dari gambar.



Gambar1.1Bag of visual words tidak memberikan informasi geometris.
(Sumber gambar: Slide lecturer Computer Vision 2013, University of Oxford)

Pada (Penatti et al., 2014) mengajukan *visual word spatial arrangement* dimana ia memberikan informasi geometris *visual words* dari gambar. Pada gambar jenis tertentu misalnya, *visual word spatial arrangement* akan memberikan informasi *visual word* tertentu memiliki kecenderungan berada pada bagian tertentu, semisal bagian kanan atas. Pada Gambar 1.2 merupakan keterangan karakteristik dari *bag of visual words* sebelumnya yaitu dengan hilangnya informasi *spatial* pada *bag of visual words* jenis gambar yang berbeda dapat memiliki *bag of visual words* yang mirip. Pada Gambar 1.2 dibawah ini ditunjukkan 3 gambar dengan dibawahnya adalah histogram dari jumlahan *word* yang dihasilkan oleh *bag of visual words*. Dapat dilihat bahwa jenis gambar yang berbeda dapat memiliki histogram *bag of visual words* yang mirip, yaitu pada 3 gambar tersebut memiliki histogram yang berbentuk identik. Terlihat di bagian jumlah pada histogram di sekitar nilai 60 memiliki ketinggian yang hampir sama. Dengan bagian lainnya mengikuti relatif rendah yang sama. Dengan kata lain histogram ketiga gambar

tersebut mirip. Padahal ketiga gambar tersebut merupakan gambar dengan jenis yang berbeda.

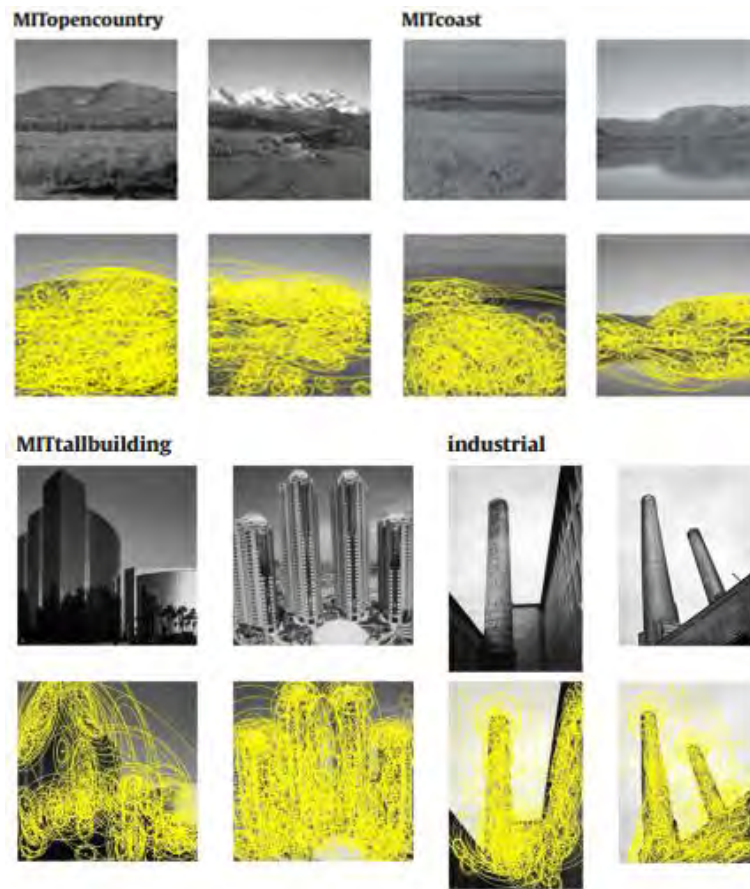


Gambar 1.2 Dengan tidak adanya informasi geometris pada bag of visual words jenis gambar yang berbeda dapat memiliki bag of visual words yang mirip (Sumber gambar: Penatti et al., 2014)

Pada *visual word spatial arrangement* (Penatti et al., 2014) *local image descriptor* diekstraksi pada *interest point*. Sekumpulan *interest point* yang digunakan untuk mengekstraksi *local image descriptor* memiliki potensi sekumpulan titik tersebut tidak dapat diandalkan (Simonyan et al., 2013). Pada hasil uji coba *visual word spatial arrangement*, *visual word spatial arrangement* kebanyakan kurang dapat membedakan gambar MITopencountry dengan gambar MITcoast dimana keduanya banyak terdapat langit cerah. Dan *visual word spatial arrangement* juga kurang dapat membedakan gambar MITtallbuilding dengan gambar industrial dimana banyak gambar struktur bangunan tinggi (Penatti et al., 2014). Penjelasannya ditunjukkan pada Gambar 1.3 dimana adalah merupakan visualisasi hasil detektor dari WSA. Sebagai contoh gambar *interest point* pada penjelasan sebelumnya dari MITopencountry, MITcoast, MITtallbuilding dan industrial. Hasil detektor yang berupa *interest point* digambarkan dengan garis-garis yang menghubungkan antar *interest point* tersebut.

Pada tesis ini diajukan metode modifikasi dari *visual word spatial arrangement* yaitu modifikasi pada bagian ekstraksi fitur *local descriptor* dengan tidak menggunakan *interest point* karena ekstraksi fitur *local descriptor* dengan

interest point berpotensi menghasilkan sekumpulan titik yang tidak dapat diandalkan.



Gambar 1.3 Interest point pada MITopencountry dan MITcoast serta MITtallbuilding dan industrial pada visual *word* spatial arrangement (Sumber gambar: Penatti et al., 2014)

Oleh karena itu pada tesis ini ekstraksi fitur *local descriptor* dengan *interest point* diganti dengan ekstraksi fitur *local descriptor* yang dihitung pada bagian komponen yang saling berdekatan atau *densely* pada keseluruhan bagian gambar. Hasil metode yang diajukan kemudian juga akan diimplementasikan sebagai salah satu fitur pada bidang tanaman yaitu *klasifikasi* jenis bunga. Pertama dilakukan ekstraksi fitur setelah dilakukan ekstraksi fitur maka untuk mendapatkan *word* dilakukan proses encoding dan *pooling* untuk menjadikannya sebagai satu fitur vektor. Dengan kata lain menggabungkan fitur hasil metode yang diajukan dengan fitur dari hasil ekstraksi fitur warna dan tekstur pada klasifikasi jenis bunga.

1.2 Perumusan Masalah

Permasalahan – permasalahan yang dikemukakan untuk dipecahkan dalam tesis ini adalah sebagai berikut:

1. Bagaimana melakukan *dense visual word spatial arrangement* pada citra umum?
2. Bagaimana melakukan *dense visual word spatial arrangement* pada citra bunga?
3. Bagaimana melakukan klasifikasi bunga secara otomatis dengan menggunakan kombinasi ekstraksi *fitur* warna, tekstur dan *visual word* pada citra bunga?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah perbaikan *visual word spatial arrangement* kemudian menerapkannya dalam data *real* yaitubersama ekstrasi fitur warna dan tekstur pada jendela objek untuk klasifikasi citra jenis bunga secara otomatis

Manfaat penelitian ini adalah dihasilkannya metode perbaikan *visual word spatial arrangement* dan model klasifikasi jenis bunga yang dapat mengenali jenis bunga secara otomatis serta jendela objek yang dapat digunakan untuk memperkirakan posisi objek bunga pada citra

1.4 Batasan Masalah

Dalam tesis ini, batasan masalah yang dibahas diuraikan sebagai berikut:

1. Visual words yang digunakan yaitu 2, 4 dan 6 words
2. Dataset yang digunakan untuk *visual word* adalah dataset 15-Scenes
3. Dataset yang digunakan adalah dataset real 30 jenis bunga dengan jumlah keseluruhan 600 citra
4. Perangkat lunak yang digunakan adalah Matlab 2012a sebagai programming tool utama, kemudian VLfeat 0.97 yang digunakan sebagai library untuk *SIFT* dan Weka 6 untuk evaluasi performa dari metode

1.5 Kontribusi

Kontribusi penulis dalam penelitian ini antara lain:

1. Metode *visual word* baru untuk mengenali konsep gambar
2. Penggabungan fitur deteksi objek dengan kombinasi warna, tekstur dan *visual word* dengan *fitur* vektor yang relatif sedikit.

BAB 2

KAJIAN PUSTAKA

2.1 Bag of Visual Word, dan Visual Word Spatial Arrangement

Metode *bag of visual words* (BoVW) dan *visual word spatial arrangement* (WSA) merupakan metode yang memberikan ciri kepada gambar dengan memberikan *word* kepada gambar tersebut (Penatti et al., 2014). Secara sederhana dapat dikatakan ciri dari gambar dapat diketahui berdasarkan *word*nya.

Suatu gambar akan memiliki *word* yang beragam dari satu jenis gambar dengan jenis gambar lainnya. Dengan kata lain jenis gambar tertentu dapat dicirikan dengan *word* yang terdapat pada gambar tersebut. Memang tidak secara langsung pada gambar, dapat terlihat cirinya walaupun gambar tersebut telah diketahui *word*nya. Secara sekilas terlihat hanya berupa titik-titik *word* pada gambar tersebut. Terlihat seperti titik-titik *word* yang berbeda dari satu gambar dengan gambar yang lain. Tetapi apabila dengan adanya banyak gambar yang memiliki jenis yang sama kemudian ditambah metode pembelajaran akan dapat mengenali gambar tersebut walaupun tentu saja ini bergantung terhadap *word* yang digunakan.



Gambar 2.1. Hutan yang dilihat dari atas

Metode *word* ini unggul apabila gambar yang akan dikenali merupakan gambar yang tidak teratur dalam jenisnya. Bukan gambar yang dapat dikenali hanya

dengan menggunakan ekstraksi fitur yang populer saja seperti ekstraksi fitur tekstur, fitur warna ataupun fitur bentuk



Gambar 2.2. Hutan yang memiliki sungai

Contohnya adalah apabila gambar yang dikenali adalah hutan dimana gambar hutan tersebut banyak orang yang berlalu-lalang, atau di dalamnya ada orang yang sedang berkemah. Bisa juga hutan tersebut sedang hujan atau hutan terbakar sehingga terlihat asap. Jadi gambar yang dikenali tidak berupa gambar hutan yang sama. Bisa juga terdapat sungai di dalam hutan tersebut. Dapat dilihat beberapa gambar hutan sebagaimana maksud gambar yang tidak teratur seperti pada Gambar 2.1, Gambar 2.2, Gambar 2.3, Gambar 2.4 dan Gambar 2.5.

Pada Gambar 2.1 merupakan gambar hutan yang dilihat dari atas. Terlihat daun-daunnya yang rimbun secara berkelompok. Gambar 2.2 merupakan gambar hutan yang memiliki sungai. Tidak seperti pada Gambar 2.1 gambar hutan ini terlihat jelas sungai yang mengalir dengan pepohonannya yang dapat terlihat di kejauhan. Pada Gambar 2.3 adalah gambar hutan pada musim salju. Dengan gambarnya yang lebih terlihat batang kayunya dibanding pada gambar 2.1. Perbedaannya juga musim salju memberikan warna putih pada beberapa daerah lingkungan hutan tersebut. Gambar 2.4 adalah hutan dengan air terjun. Selain hutan ini memiliki sungai sebagaimana pada Gambar 2.2 terdapat air terjun yang dapat terlihat di kejauhan. Di sekelilingnya juga terdapat pepohonan dengan skala yang

lebih besar dari gambar-gambar sebelumnya. Sebagai contoh untuk gambar hutan yang terakhir yaitu Gambar 2.5. Dapat dilihat pada gambar tersebut terdapat hewan yang melintasi hutan. Sekumpulan hewan yang bergerak dari sebelah kiri gambar menuju sebelah kanan. Pada gambar ini berbeda dari gambar-gambar sebelumnya yang merupakan gambar hutan dengan adanya sungai dan air terjun.



Gambar 2.3. Hutan pada musim salju

Jadi gambar yang tidak teratur ini dapat direpresentasikan oleh contoh gambar-gambar yang dijelaskan sebelumnya.



Gambar 2.4. Hutan dengan air terjun

Pada gambar – gambar hutan tersebut dapat ada objek-objek di dalamnya semisal air terjun atau juga gambar hutan yang dilihat dari kejauhan dari atas.



Gambar 2.5. Hutan dengan hewan di dalamnya

Contoh gambar lainnya yaitu gambar gunung yang ditunjukkan pada Gambar 2.6, Gambar 2.7, Gambar 2.8, Gambar 2.9, Gambar 2.10, Gambar 2.11 dan Gambar 2.12. Pada gambar 2.6 adalah gambar gunung yang bersalju dengan terdapat pendaki gunung pada gambar tersebut. Yaitu tiga orang pendaki gunung. Di kejauhan juga terlihat beberapa pepohonan di sekitar gunung tersebut. Gambar 2.8 merupakan gambar gunung dengan bulan yang terdapat pada bagian kiri atas gunung tersebut. Pada gambar ini terdapat objek bulan yang berbeda dari gambar gunung sebelumnya



Gambar 2.6. Gambar gunung dengan pendaki gunung

Pada gambar 2.7 merupakan gambar gunung yang memiliki asap. Terlihat pada bagian atas gambar yaitu asap berwarna putih yang mengelilingi gunung.



Gambar 2.7. Gunung dengan asapnya



Gambar 2.8. Gunung dengan langitnya terdapat objek terang

Di gambar 2.9 merupakan gambar gunung dengan sekeliling bagian bawahnya merupakan sungai. Terlihat gunung yang menjulang tinggi dan sungai yang lebar pada bagian bawahnya.

Pada gambar 2.10 adalah gambar gunung yang juga berbeda dari gambar sebelumnya. Terdapat sungai yang mengalir yang terlihat pada bagian bawah gambar dan disekelilingnya ada bangunan-bangunan. Bangunan-bangunan ini terlihat berada di samping kiri dan kanan dari aliran sungai.



Gambar 2.9. Gunung dengan sungainya



Gambar 2.10. Gunung dengan sungai dan terdapat bangunan disekitarnya

Berbeda lagi dengan gunung yang ditunjukkan pada Gambar 2.11. Pada gambar ini puncak gunung tidak terlihat. Bulan terlihat berada di langit malam sebelah bagian tengah agak ke kiri atas dari gambar. Tampak di kejauhan pemukiman yang berada di sekitar bagian bawah dari gunung.

Pada gambar 2.12 adalah gambar gunung-gunung yang terlihat jelas pendaki gunung. Pendaki gunung ini tampak jelas sehingga skalanya cukup besar terlihat pada gambar. Pada kejauhan masih terlihat puncak-puncak dari gunung dan puncak gunung-gunung ini terlihat bersalju.



Gambar 2.11. Gunung dengan perumahan disekitarnya dan cahaya bulan



Gambar 2.12. Gunung dengan pendaki yang menuruni gunung

Pada penjelasan sebelumnya adalah penjelasan gambar-gambar yang tidak beraturan tetapi masih dalam satu jenis. Dimana metode *word* percaya diri dalam mengenali gambar tersebut.

Gambar berbeda yang memiliki jenis seperti ini juga ditunjukkan pada Gambar 2.13, Gambar 2.14, Gambar 2.15, Gambar 2.16, Gambar 2.17 dan Gambar 2.18. Gambar 2.13 sampai Gambar 2.17 merupakan jenis gambar pertokoan dan Gambar 2.18 merupakan jenis gambar perindustrian. Dapat terlihat berbagai macam objek yang berbeda pada gambar tersebut tetapi masih dalam satu kategori yaitu gambar pertokoan.



Gambar 2.13.Toko dengan penjualan topi



Gambar 2.14.Toko dengan penjualan sepatu



Gambar 2.15.Toko dengan buku-bukunya



Gambar 2.16.Toko dengan orang di depan penjualan

Berbeda dengan apabila ingin mengenali jenis gambar yang teratur semisal gambar permukaan daun dimana daun sudah diletakkan sedemikian rupa sehingga struktur daunnya terlihat. Maka jenis daun dapat dikenali dengan semisal melakukan ekstraksi fitur yang populer. Seperti ekstraksi fitur tekstur dan fitur bentuk pada daun tersebut. Atau juga semisal pengenalan jenis ikan dimana ikan dengan jenis yang berbeda diletakkan pada tempat tertentu. Sehingga dapat dikenali dengan menggunakan ekstraksi fitur warna, fitur tekstur dan fitur bentuk



Gambar 2.17. Toko dengan penjualan yang tertata di rak



Gambar 2.18. Perindustrian dengan cerobong asapnya

Untuk contoh gambar yang teratur ini dapat dilihat pada Gambar 2.19 yang merupakan gambar jenis ikan lele dan Gambar 2.20 yang merupakan contoh gambar mujair. Dapat dilihat pada gambar contoh jenis ikan lele yaitu pada Gambar 2.19 pada gambar tersebut objek diletakkan pada seperti wadah dengan latar belakang wadah tersebut berwarna berbeda dari objek ikan. Ikan berada pada sekitar bagian tengah dari gambar dengan warnanya yang kehitaman. Untuk gambar yang teratur maka pada sampel-sampel gambar ikan lele berikutnya memiliki karakteristik penempatan yang mirip antar satu gambar dengan gambar yang

lainnya. Jadi ikan lele berada pada bagian tengah gambar dan latar belakang yang relatif berwarna putih. Begitu juga pada gambar jenis ikan mujair.



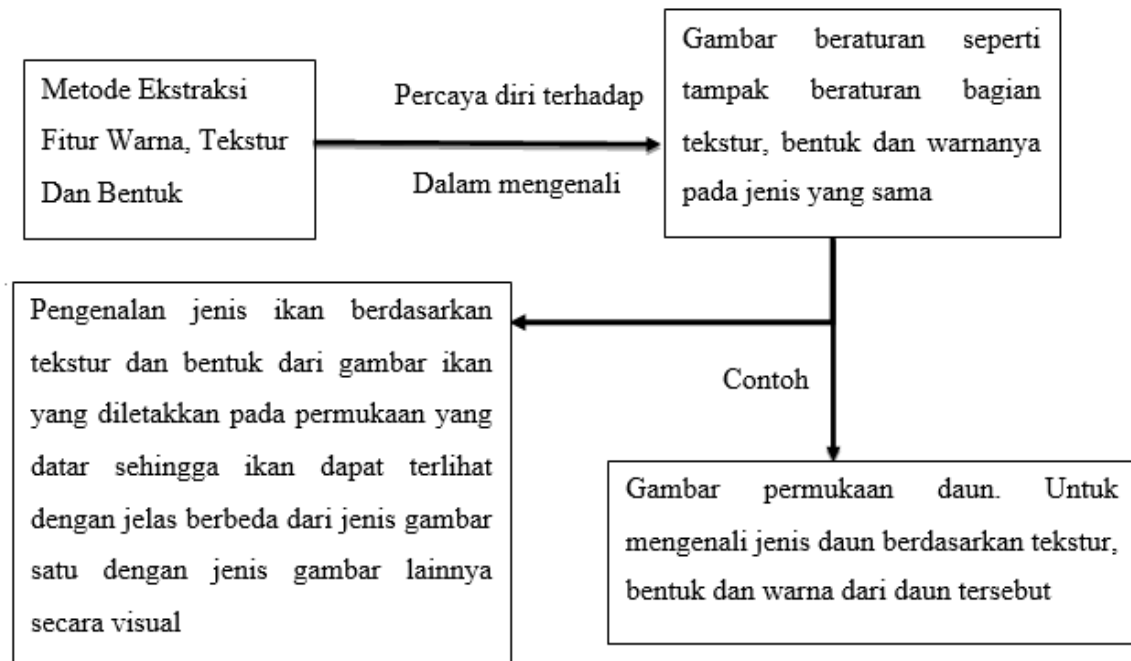
Gambar 2.19. Contoh gambar jenis ikan lele



Gambar 2.20. Contoh gambar jenis ikan Mujair

Gambar 2.21 menjelaskan mengenai kepopuleran menggunakan ekstraksi fitur dalam mengenali jenis gambar.

Secara ringkas kedua metode ini yaitu pengenalan jenis gambar dengan menggunakan *word* dan pengenalan jenis gambar dengan menggunakan ekstraksi fitur seperti ekstraksi fitur warna, ekstraksi fitur tekstur dan ekstraksi fitur bentuk dapat dilihat pada Gambar 2.21 dan Gambar 2.22. Gambar 2.21 merupakan diagram untuk ekstraksi fitur dalam mengenali jenis gambar sedangkan Gambar 2.22 merupakan diagram *word* dalam mengenali jenis gambar. Pada Gambar 2.22 juga diberikan contohnya.



Gambar 2.21. Gambar diagram ekstraksi fitur dalam mengenali jenis gambar

2.2 Bag of Visual Word

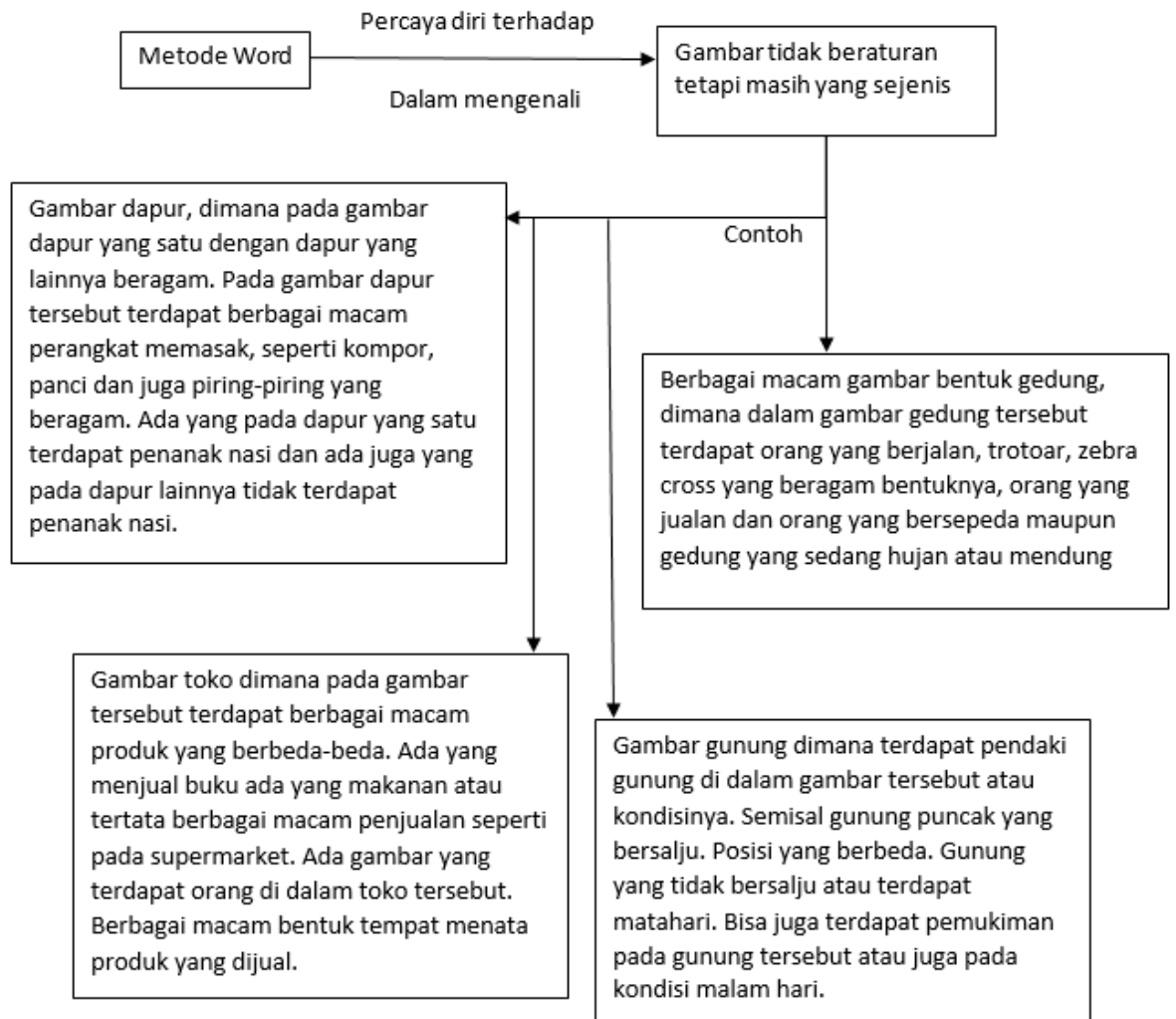
Metode bag of visual word ditunjukkan pada Gambar 2.23. Pertama dilakukan ekstraksi fitur deskriptor yang ditunjukkan oleh kotak yang paling kiri. Kemudian arah panah menunjuk ke kotak *encoding*. Menunjukkan hasil fitur deskriptor sebelumnya diproses kemudian dengan proses *encoding*. Kotak terakhir sebelah kanan adalah proses *pooling* dimana memproses hasil *encoding* sebelumnya menjadi satu fitur vektor. Bagian kanan sendiri adalah selesai yang menunjukkan hasil fitur vektor dari *pooling* inilah yang digunakan untuk proses *learning*.

Pada Gambar 2.4 adalah gambar matrix untuk proses *coding* dan *pooling*. Dengan c_1, c_m, c_M adalah *word* ke 1, *word* ke m , dan *word* ke M . x_1, x_j , dan x_N adalah *instance* ke 1, *instance* ke j dan *instance* ke N .

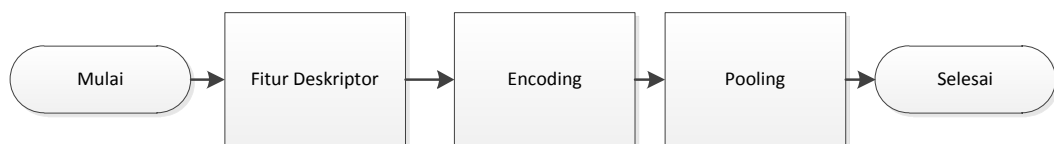
Perhitungan untuk nilai lambda menggunakan persamaan (2) di bawah ini

$$\alpha_{m,j} = 1 \text{ iff } m = \arg \min_{k \in \{1, \dots, M\}} \|x_j - c_k\|_2^2 \quad (2)$$

Dengan c_1, c_m, c_M adalah *word* ke 1, *word* ke m , dan *word* ke M . x_1, x_j , dan x_N adalah *instance* ke 1, *instance* ke j dan *instance* ke N .



Gambar 2.22. Diagram kepopuleran metode *word* dalam mengenali jenis gambar



Gambar 2.23. Diagram alir dari bag of visual *word*

$$\begin{array}{c}
\begin{array}{ccc} x_1 & x_j & x_N \end{array} \\
\begin{array}{c} c_1 \\ \vdots \\ c_m \\ \vdots \\ c_M \end{array} \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \Rightarrow g: \text{pooling} \\
\Downarrow \\
f: \text{coding}
\end{array}$$

Gambar 2.24. Matrix untuk proses coding dan pooling

2.3 Visual Word Spatial Arrangement

Pada bagian ini dijelaskan secara umum dari metode *visual word spatial arrangement* (Penatti et al., 2014). *Local descriptor* didapat dengan *interest point detector*, kemudian dilakukan *encoding*, setelah itu pada setiap *local descriptor* pada citra dihitung berapa banyak *word* sesuai *local descriptor* tersebut pada setiap kuadran. Kemudian setelah selesai dihitung untuk semua *local descriptor*,, dilakukan normalisasi pada setiap hasil perhitungan *word* tersebut. Fitur *vector* yang dihasilkan sebanyak $4W$ dengan W adalah jumlah *word*.

2.3.1. Interest Point

Interest point merupakan titik-titik yang merepresentasikan daerah penting dari permukaan gambar. *Interest point* yang didapat dari gambar dihasilkan oleh detektor. Detektor mencari titik-titik mana yang merupakan bagian penting dari gambar sehingga nantinya dari titik-titik ini dihasilkan sesuatu yang merepresentasikan gambar. Sesuatu ini adalah deskriptor. Dari titik-titik inilah deskriptor dihasilkan. Deskriptor merupakan representasi permukaan tampilan gambar yang dikemas secara padat dan handal.

2.3.2. Detektor Scale-Invariant Feature Transform

Titik-titik atau *keypoint* SIFT merupakan daerah yang berbentuk melingkar dengan sebuah orientasi. Titik-titik tersebut dideskripsikan dengan empat parameter

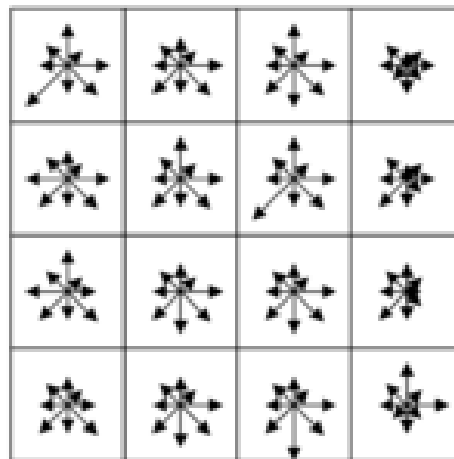
yaitu pusat *keypoint* x dan y , skala atau disebut juga radius dari daerah *keypoint* tersebut dan orientasinya.

SIFT detektor mencari *keypoints* struktur gambar yang menyerupai "*blobs*" atau gumpalan. Dengan mencari *blobs* pada skala dan posisi yang beragam. Detektor SIFT adalah *invariant* terhadap translasi, rotasi dan *re scaling* dari gambar. Pada gambar di lakukan penskalaan yang berbeda-beda. Dengan setiap hasil penskalaan tersebut gambar seperti secara bertahap memiliki resolusi gambar lebih rendah dari gambar skala sebelumnya. Seperti gambar lebih *smooth*. Dari setiap gambar yang berbeda-beda skala tersebutlah dicari *blobs* nya.

Keypoints didapatkan dengan menghilangkan titik-titik yang sepertinya adalah tidak stabil. Apakah karena letaknya yang jauh dari *blob* atau letaknya ditempat pada struktur gambar dengan kontras yang rendah.

2.3.3. Deskriptor Scale-Invariant Feature Transform

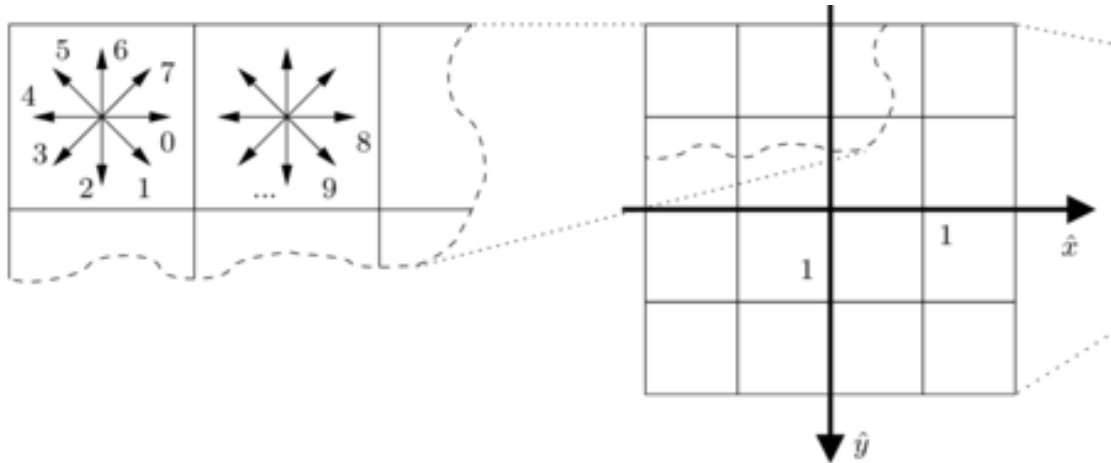
Deskriptor SIFT adalah 3-D *spatial* histogram dari gradien gambar yang mengkarakteristikan permukaan tampilan dari *keypoint*. Gambar *spatial* histogramnya dapat dilihat pada gambar 2.25 di bawah ini.



Gambar 2.25. 3D Spatial Histogram SIFT

Sedangkan koordinat histogramnya dapat dilihat pada gambar 2.26 di bawah ini. Dari histogram ini merupakan representasi dari deskriptor lokal. Setiap

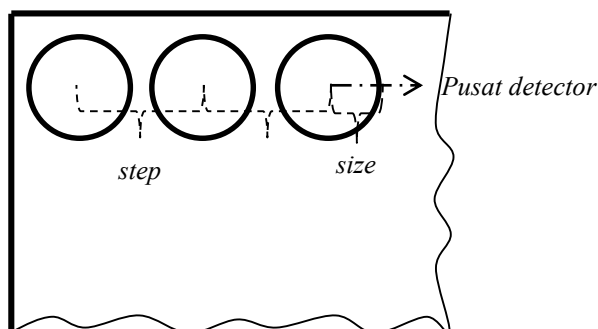
bagian dari *keypoint* memiliki 8 nilai dari setiap sub daerahnya yang dapat dilihat pada gambar adalah kotak yang kecil. Dengan jumlah keseluruhan kotak adalah 16 jadi dihasilkan 128 fitur dari SIFT. 128 fitur dari SIFT untuk setiap deskriptor lokal.



Gambar 2.26. Orientasi pada SIFT dan jumlahnya untuk setiap subblok

2.4 Dense SIFT

Pada bagian ini dijelaskan secara umum mengenai *denseSIFT*. *Dense SIFT* digunakan sebagai *point detector*. Point detector ini yang digunakan sebagai lokasi untuk dilakukannya ekstraksi fitur descriptor. Pada *dense SIFT* terdapat dua parameter *detector* yaitu *step* dan *size*. *Step* digunakan untuk setiap berapa *step* piksel pusat *detector* ini berada, sedangkan *size* digunakan sebagai ukuran jari-jari dari *detector*. Diilustrasikan pada gambar 2.3 di bawah ini



Gambar 2.27. Ilustrasi detektor dense SIFT pada potongan citra

2.5 Evaluasi

Evaluasi metode dilakukan untuk mengetahui seberapa efektif suatu metode. Pengukuran evaluasi ini dilakukan dengan menguji presisi, *recall*, dan *f-measure*

2.5.1. Presisi

Presisi merupakan persentase *instance* yang benar terlabeli sebagai positif. Pengukuran presisi ini memberikan evaluasi ketepatan hasil yang diberikan. Nilai presisi didapat dengan persamaan (3) di bawah ini

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Dengan TP adalah *true* positif dan FP adalah *false* positif.

2.5.2. Recall atau True Positif Rate (TPRate)

Recall merupakan persentase *instance* positif yang benar terklasifikasi. *Instance* positif merupakan *instance* dengan kelas yang bersangkutan. Dengan merujuk pada Tabel 2.1, nilai *TPRate* didapat dengan persamaan (4) di bawah ini

$$TPRate = \frac{TP}{TP + FN} \quad (4)$$

Dengan TP adalah *true* positif, FP adalah *false* positif dan FN adalah *false* negative.

2.5.3. F-Measure (F₁)

Pengukuran *F-Measure* ini mengintegrasikan presisi dan *recall*. *F-Measure* digunakan untuk menggabungkan presisi dan *TPRate* menjadi satu *metric*, merepresentasikan *weighted harmonic mean* antara dua *metric* tersebut. Dengan merujuk pada Tabel 2.1, nilai *F-Measure* didapat dengan persamaan (8) di bawah ini

Tabel 2.1 Confusion matrix

	Predicted positive	Predicted negative
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{TPrate}}{\text{Precision} + \text{TPrate}} \quad (8)$$

Dengan perhitungan presisi seperti pada persamaan (3) dan *TPrate* seperti pada persamaan(4).

BAB 3

METODE PENELITIAN

3.1 Rancangan Penelitian

Pada bagian ini dijelaskan rancangan penelitian secara umum. Pertama adalah pengerjaan metode yang dilakukan pada dataset citra umum dan kedua adalah pengerjaan metode yang dilakukan pada dataset citra bunga. Diagram alir rancangan penelitian secara umum dapat dilihat pada Gambar 3.1.

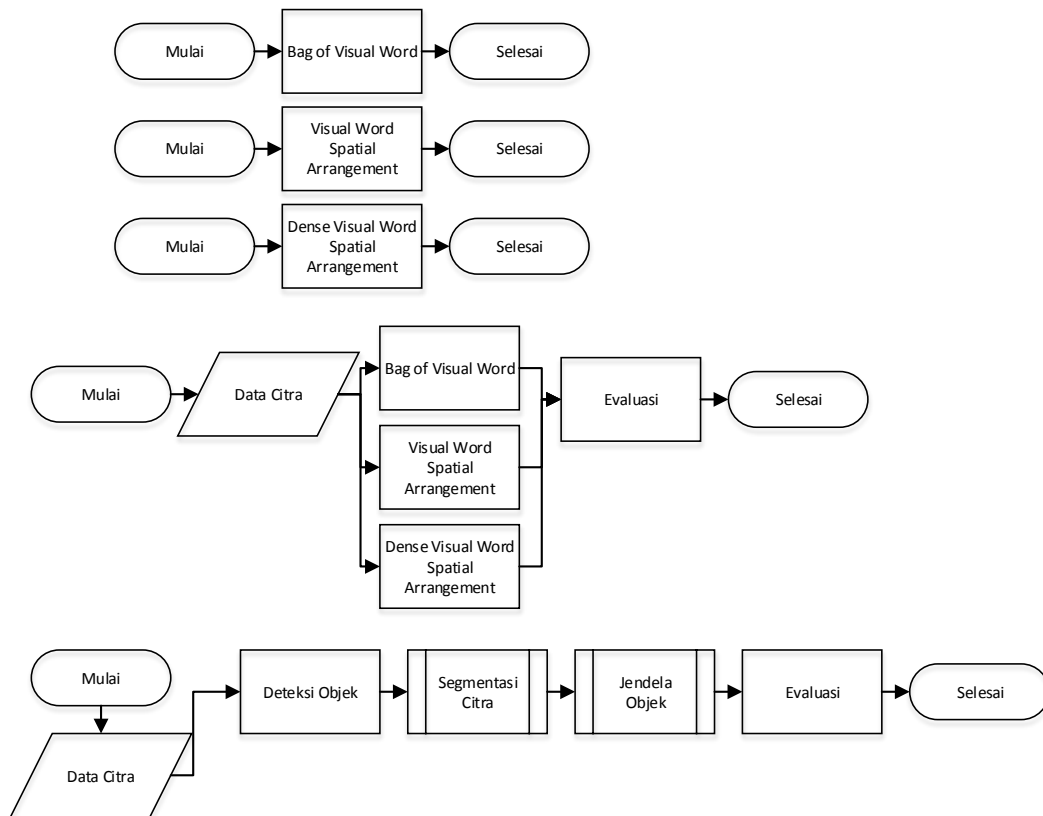
3.1.1. Pengerjaan pada dataset citra umum

Pengerjaan metode dilakukan dengan *bag of visual word*, *visual word spatial arrangement* dan metode yang diajukan yaitu *dense visual word spatial arrangement* pada dataset 15-scenes. Diagram alir dari *dense visual word spatial arrangement* dapat dilihat pada Gambar 3.5. Untuk diagram alir rancangan penelitian secara umum pertama adalah diagram pengerjaan metode. Ditunjukkan pada Gambar 3.1 bagian diagram pertama.

Terdapat tiga sub diagram dari pengerjaan metode. Pertama adalah mulai, Bag of Visual Word, selesai. Kedua adalah Mulai, Visual Word Spatial Arrangement, selesai. Dan yang ketiga adalah Mulai, Dense Visual Word Spatial Arrangement, selesai. Ketiga sub diagram ini menunjukkan dilakukannya tiga metode yaitu Bag of Visual Word, Visual Word Spatial Arrangement dan Dense Visual Word Spatial Arrangement. Untuk bagian diagram kedua dari Gambar 3.1 adalah diagram evaluasi performa dari metode secara umum.

Pertama input berupa data citra. Dari data citra ini diproses dengan Bag of Visual Word, Visual Word Spatial Arrangement dan Dense Visual Word Spatial Arrangement. Kemudian hasilnya dievaluasi. Ditunjukkan ketiga arah panah dari masing-masing metode menuju balok proses evaluasi. Untuk bagian diagram terakhir dari Gambar 3.1 adalah diagram secara umum dari penerapan metode bersama fitur warna dan tekstur serta evaluasi performa dari penerapan ini. Dimulai

dengan inputan berupa data citra dimana data citra ini dilakukan sub proses segmentasi, pengambilan jendela objek dan ekstraksi fitur warna, fitur tekstur dan *visual word*. Hasilnya kemudian di evaluasi.



Gambar 3.1 Diagram alir tahapan metodologi

3.1.2. Pengerjaan pada dataset citra bunga

Pengerjaan dilakukan pada dataset 30 jenis bunga dengan melakukan *cropping* secara otomatis pada objek bunga kemudian hasil *cropping* tersebut dilakukan ekstraksi fitur warna, tekstur dan *dense visual word spatial arrangement*. Diagram alirnya dapat dilihat pada Gambar 3.1 point ke 3.

3.2 Skenario Uji Coba

Pada bagian ini dijelaskan secara garis besar skenario uji coba. Terdapat dua bagian utama skenario yaitu bagian pertama uji coba metode yang diusulkan terhadap metode yang sudah ada. Kemudian bagian kedua uji coba metode *word* apabila disandingkan dengan metode ekstraksi fitur.

Pada skenario yang pertama yaitu uji coba metode yang diusulkan terhadap metode yang sudah ada. Tujuannya adalah untuk mengetahui hasil dari metode yang diusulkan apakah performanya lebih bagus atau tidak tanpa ada penambahan fitur apapun selain fitur *word* dari setiap metode. Jadi hanya fitur hasil *word* dari masing-masing metode yang digunakan sebagai fitur vektor. Tidak ada penggunaan fitur lain. Karena menggunakan fitur hasil *word* dari masing-masing metode maka yang pertama kali didefinisikan adalah berapa *word* yang akan digunakan.

Word yang digunakan sebagai skenario ini adalah berjumlah 2 *word*, 4 *word* dan 6 *word*. Sedangkan metode yang akan dibandingkan dengan metode yang diusulkan adalah metode Bag of Visual *Word* dan Visual *WordSpatial* Arrangement. Jadi terdapat 3 x 3 uji coba pada skenario yang pertama ini. Yaitu berjumlah 9 uji coba.

Pada skenario yang kedua yaitu uji coba metode *word* apabila disandingkan dengan metode ekstraksi fitur lain. Tujuannya adalah untuk mengetahui apakah bisa fitur dari *word* digabungkan dengan fitur dari metode ekstraksi fitur lain. Dilihat dengan bagaimana performa yang dihasilkan dari penggabungan tersebut. Apakah lebih baik dari hanya metode ekstraksi fitur yang sudah ada. Jadi pada skenario kedua ini terdapat metode ekstraksi fitur yang sudah ada dimana metode ekstraksi fitur yang sudah ada tersebut adalah metode dasar yang dibandingkan. Sedangkan untuk metode *word* yang dibandingkan adalah metode *bag of visual word*, *visual word spatial arrangement* dan *dense visual word spatial arrangement*.

Ekstraksi fitur untuk metode dasar adalah ekstraksi fitur warna dan ekstraksi fitur tekstur. Ekstraksi fitur warna menggunakan nilai a^*b^* dari ruang warna Lab, nilai H,S,dan V dari ruang warna HSV dan ekstraksi fitur tekstur menggunakan nilai kontras, energi, homogenitas dan korelasi dari GLCM. *Word* yang digunakan berjumlah 2 *word*. Jadi pada skenario kedua ini terdapat 4 kali uji coba.

3.3 Langkah-langkah Penelitian

Pada bagian ini dijelaskan mengenai langkah-langkah penelitian yang dilakukan. Dengan bagian pertama adalah mengenai pemilihan dataset, pengelompokan secara manual jenis dataset dan dilanjutkan dengan pembuatan *visual dictionary*.

3.3.1. Pemilihan dataset

Pada tesis ini digunakan dua jenis dataset yaitu dataset untuk menguji metode yang diajukan sebagaimana yang digunakan oleh (Penatti et al., 2014). Dataset tersebut adalah dataset 15-*scenes* dan untuk penerapannya pada data *real* yaitu menggunakan dataset 30 jenis bunga berjumlah 600 hasil foto kamera *handphone*. Untuk dataset 15-*scenes*, terdiri dari 4485 citra, yang dibagi menjadi 15 kategori. Contoh dataset ini dapat dilihat pada Gambar 3.2. Dataset pada Gambar 3.2 ini merupakan dataset citra umum. Dimana pada contoh di bawah terdapat 3 kali 7 sampel gambar. Yaitu 21 sampel gambar. Dengan gambar ada yang berupa pegunungan seperti ditunjukkan pada gambar pojok kiri atas, kemudian ada gambar bangunan, toko dan perindustrian. Dapat dilihat secara lebih lanjut pada Gambar 3.2 di bawah ini.



Gambar 3.2 Contoh citra dari dataset 15-scenes

Dataset citra umum ini memiliki 15 jenis kategori yaitu bedroom, CALsuburb, industrial, kitchen, livingroom, MITcoast, MITforest, MITHighway, MITinsidecity, MITmountain, MITopencountry, MITstreet, MITtallbuilding,

PARoffice dan store. Penamaan label dan jenis untuk dataset ini dapat dilihat pada Tabel 3.1 dibawah.

Label	Jenis	Label	Jenis	Label	Jenis
1	bedroom	6	MITcoast	11	MITopencountry
2	CALsuburb	7	MITforest	12	MITstreet
3	industrial	8	MIThighway	13	MITtallbuilding
4	kitchen	9	MITinsidecity	14	PARoffice
5	livingroom	10	MITmountain	15	Store

Sedangkan untuk dataset citra jenis bunga, citra di foto dengan beragam kondisi waktu. Beberapa citra yang mewakili jenis bunga dapat dilihat pada Gambar 3.3. Dapat dilihat berbagai macam warna, bentuk dan pencahayaan yang berbeda pada gambar dataset bunga ini. Sebagai contoh pada Gambar 3.3 bunga yang berada pada pojok kiri atas, memiliki warna bunga putih dengan benang sari pada bagian tengahnya berwarna kuning. Pada sekelilingnya terdapat daun-daun yang berbentuk seperti batangan-batangan. Terlihat juga pada gambar di pojok kiri bawah pencahayaan yang berbeda dari gambar-gambar lainnya. Ada juga yang bagian bunga lebih kecil dari keseluruhan bagian gambarnya seperti pada gambar nomer 13 dari Gambar 3.3 ini.

Kemiripan warna bunga juga dapat ditemui pada beberapa gambar seperti pada gambar 10 dan gambar 18. Warna bunga tersebut dominan sama berwarna ungu. Ada juga bunga yang memiliki warna yang sama putih seperti pada gambar nomer 2, gambar nomer 5, gambar nomer 14 dan gambar nomer 24. Warna bunga yang unik seperti bagian tepinya berbeda warna dapat ditemui pada gambar nomer 20. Gambar bunga tersebut berwarna merah dengan tepinya berwarna kuning. Detail sampel setiap kelas dari dataset bunga ini dapat dilihat pada Gambar 3.3.

Pada dataset bunga ini satu objek dapat difoto lebih dari satu dengan posisi, rotasi, skala objek yang berbeda. Pengambilan objek ada yang pagi hari, siang hari dan sore hari. Dari berbagai macam kondisi ini dihasilkan citra jenis bunga dengan skala, pencahayaan, bayangan, kecerahan yang beragam.



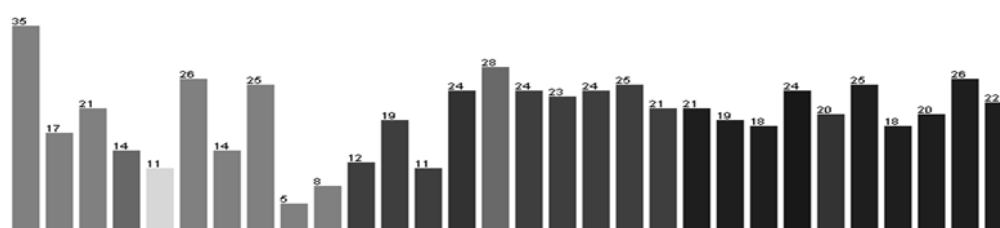
Gambar 3.3 Contoh dari dataset citra 30 jenis bunga yang digunakan beserta label jenisnya

Pada Tabel 3.2 adalah pendekatan nama jenis dari dataset bunga. Keseluruhan jenis bunga berjumlah 30 jenis. Bunga-bunga tersebut adalah bunga bawang sebrang, bunga ceplok piring, bunga delapan dewa, bunga *chamaecrista fasciculata*, bunga sepatu putih, bunga batavia, bunga bugenvil merah muda, bunga kamboja, bunga bugenvil putih, bunga kupu-kupu, bunga sepatu merah, bunga rolia pink, bunga rumput, bunga melati, bunga soka kuning dan merah, bunga dadap merah, bunga mentega, bunga sepatu kuning, bunga merak, bunga *widelia biflora*, bunga iris kuning, bunga lili, bunga tapak dara, bunga putri malu, bunga *pseuderanthemum reticulatum*, bunga tomat, bunga biduran, bunga kamboja jepang dan bunga kertas. Untuk keterangan jenis bunga bersama labelnya dapat dilihat lebih lanjut pada Tabel 3.2 di bawah ini.

Tabel 3.2 Label dataset citra 30 jenis bunga digunakan beserta nama jenisnya

Label	Jenis
1	Bunga bawang sebrang (<i>Zephyranthes candida</i>)
2	Bunga ceplok piring
3	Bunga delapan dewa (<i>Euphorbia</i>)
4	Chamaecrista fasciculata
5	Bunga sepatu putih (<i>Hibiscus rosa-sinensis</i> 'white')
6	Bunga batavia (<i>Jatropha integerrima</i>)
7	Bugenvil merah muda(<i>Bougainvillea spectabilis</i> 'pink')
8	Bunga kamboja
9	Bugenvil putih
10	Bunga kupu-kupu (<i>Bauhinia</i>)
11	Bunga sepatu merah (<i>Hibiscus rosa-sinensis</i> 'red')
12	Rolia pink
13	Bunga rumput
14	Bunga melati
15	Bunga soka kuning (<i>Ixora</i> yellow)
16	Bunga soka merah (<i>Ixora</i> red)
17	Bunga dadap merah
18	Bunga mentega
19	Bunga sepatu kuning (<i>Hibiscus rosa-sinensis</i> 'yellow')
20	Bunga merak (<i>Caesalpine pulcherima</i>)
21	Widelia biflora
22	Bunga iris kuning (<i>Neomarica longifolia</i>)
23	Bunga lili
24	Bunga tapak dara
25	Bunga putri malu
26	Pseuderanthemum reticulatum
27	Bunga tomat
28	Bunga biduran
29	Bunga kamboja jepang
30	Bunga kertas (<i>zinnia elegans</i>)

Visualisasi histogram jumlah dataset dapat dilihat pada Gambar 3.4. Secara berurutan dimulai dengan jenis 1 sebelah paling kiri histogram, kemudian sebelah kanannya jenis 2 dan seterusnya.



Gambar 3.4Histogram jumlah dan jenis citra bunga.

3.3.2. Pengelompokan secara manual jenis dataset

Pada dataset *15-scenes* masing-masing jenis gambar dikelompokkan pada *folder* yang berbeda sesuai jenis gambar tersebut, begitu juga dengan dataset 30 jenis bunga masing-masing jenis bunga juga dikelompokkan pada *folder* berbeda sesuai jenis bunga.

3.3.3. Visual Dictionary

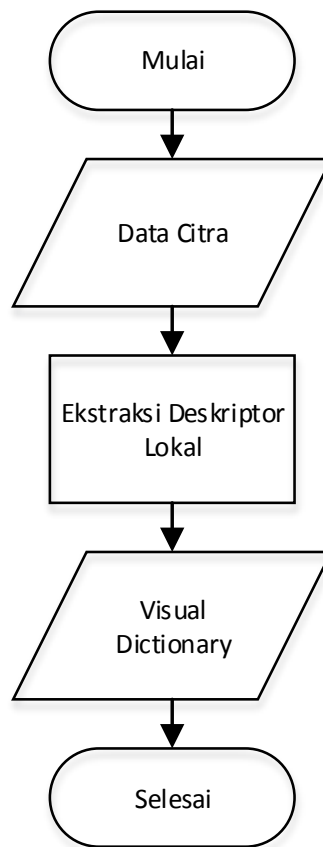
Pada dataset *15 scenes*, untuk setiap kategori diambil secara acak 30 citra, jadi total citra yang menjadi sampel untuk membuat *visual dictionary* ini adalah $15 \times 30 = 450$ citra. Kemudian pada masing-masing citra tersebut dilakukan ekstraksi local descriptor dengan menggunakan *dense SIFT*. Lalu dipilih secara acak sebanyak *word* yang akan digunakan pada *fitur vector* hasil keseluruhan dari ekstraksi *local descriptor*.

Diagram alirnya dapat dilihat pada Gambar 3.5. Dimulai input berupa data citra yang direpresentasikan dengan bentuk trapesium. Kemudian data citra tersebut diproses dengan proses ekstraksi deskriptor lokal yang direpresentasikan dengan bentuk balok. Hasil akhirnya berupa visual dictionary yang direpresentasikan dengan bentuk trapesium.

3.3.4. Dense Visual Word Spatial Arrangement

Setiap citra pada masing-masing kategori diekstraksi *local deskriptornya* dengan menggunakan *dense sift*, kemudian setiap *local descriptor* dihitung jaraknya menggunakan dengan *fitur vector visual dictionary* dan hasil terdekat dicatat sebagai *word* dari *local descriptor*. lalu pada setiap *local descriptor* dihitung berapa banyak *word* tersebut untuk setiap *kuadran*, setelah selesai hasilnya *dinormalisasi*.

Diagram alir tahapan metode yang diajukan ditunjukkan pada Gambar 3.6. Pertama dimulai dengan input berupa input citra dan visual *dictionary* yang ditandai dengan bentuk trapesium. Artinya adalah merupakan data. Kemudian pada input citra dilakukan proses *dense sift*.

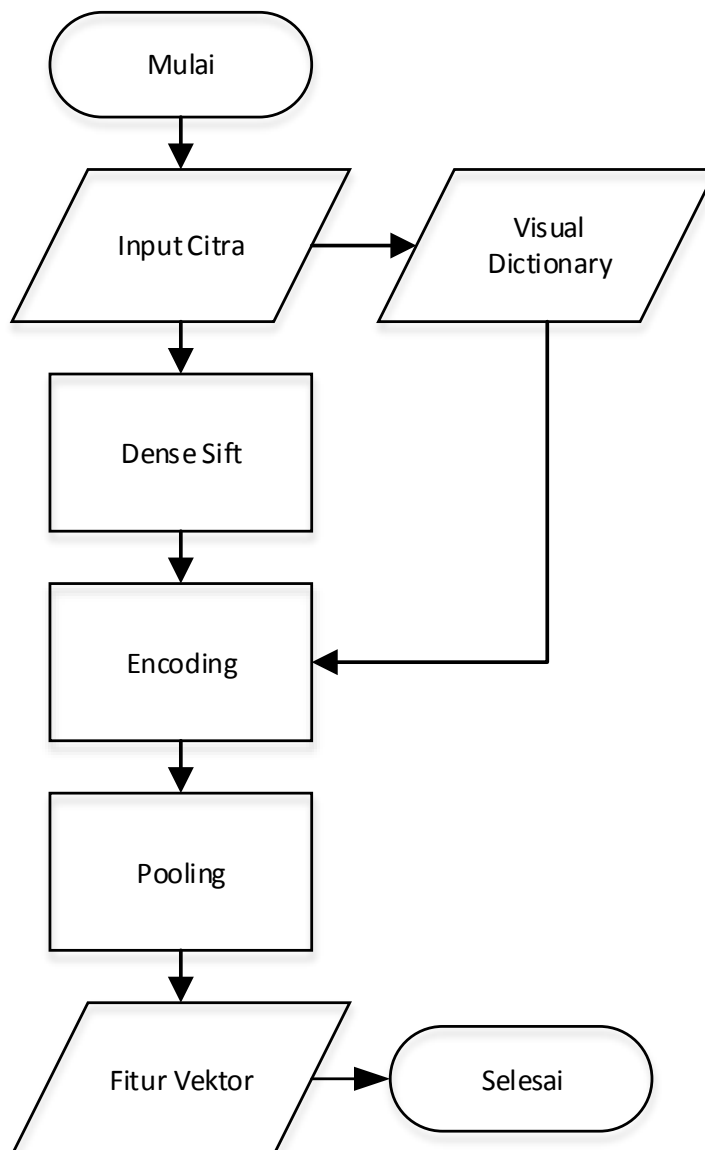


Gambar 3.5 Diagram alir tahapan membuat visual dictionary

Hasil dari *dense sift* bersama dengan visual *dictionary* ini kemudian di proses selanjutnya dengan proses yang bernama proses *encoding*. Setelah proses *encoding* ini dilakukan proses *pooling*. Hasil dari *pooling* berupa fitur vektor dimana pada diagram ditunjukkan dengan bentuk trapesium.

3.3.5. Segmentasi Citra

Segmentasi citra dilakukan pada dataset 30 jenis bunga. Pertama pada input citra dilakukan *mean filter* kemudian *clustering K-means* dengan dua *cluster*. Hasilnya kemudian dilakukan operasi *morfologi closing* untuk mendapatkan citra jendela objek. Pertama input berupa data input citra ditunjukkan dengan bentuk trapesium. Dari input citra ini diproses *clustering k-means* dengan dilakukan *mean filter* terlebih dahulu pada input citra dan inisialisasi menggunakan *k-means++*. Hasilnya berupa citra *region*.



Gambar 3.6 Diagram alir tahapan metode yang diajukan

Data citra *region* ini diproses lanjut dengan operasi *morfologi closing*. Kemudian untuk mengisi bagian daerah segmentasi dilakukan operasi *flood fill*. Dari hasil *flood fill* ini dilakukan operasi jendela objek untuk mendapatkan objek dari citra hasil segmentasi. Hasilnya berupa citra jendela objek. Diagram alir proses *segmentasi* dapat dilihat pada Gambar 3.7.

Segmentasi pada gambar akan membagi gambar menjadi daerah - daerah yang memiliki sifat yang berbeda antar daerah tersebut. Piksel-piksel pada gambar yang memiliki karakteristik yang mirip akan berada pada satu daerah sedangkan

piksel-piksel yang memiliki karakteristik yang berbeda akan berada pada daerah yang berbeda ditunjukkan dengan hasil segmentasi. Pada pengenalan objek bunga ini segmentasi bertujuan untuk memisahkan objek bunga dengan backgroundnya. jadi akan didapat piksel-piksel objek bunga saja. Dengan mendapatkan piksel-piksel objek bunga maka pengambilan ekstraksi fitur akan secara baik mendapatkan hanya objek bunga tersebut dengan beberapa *noise*. Sehingga ekstraksi fitur akan lebih fokus. Hal ini dikarenakan ekstraksi fitur yang digunakan seperti fitur warna akan mengambil warna dari keseluruhan piksel, jika hanya objek bunga saja maka pengambilan fitur warna akan lebih fokus. Begitu juga dengan menggunakan fitur tekstur.

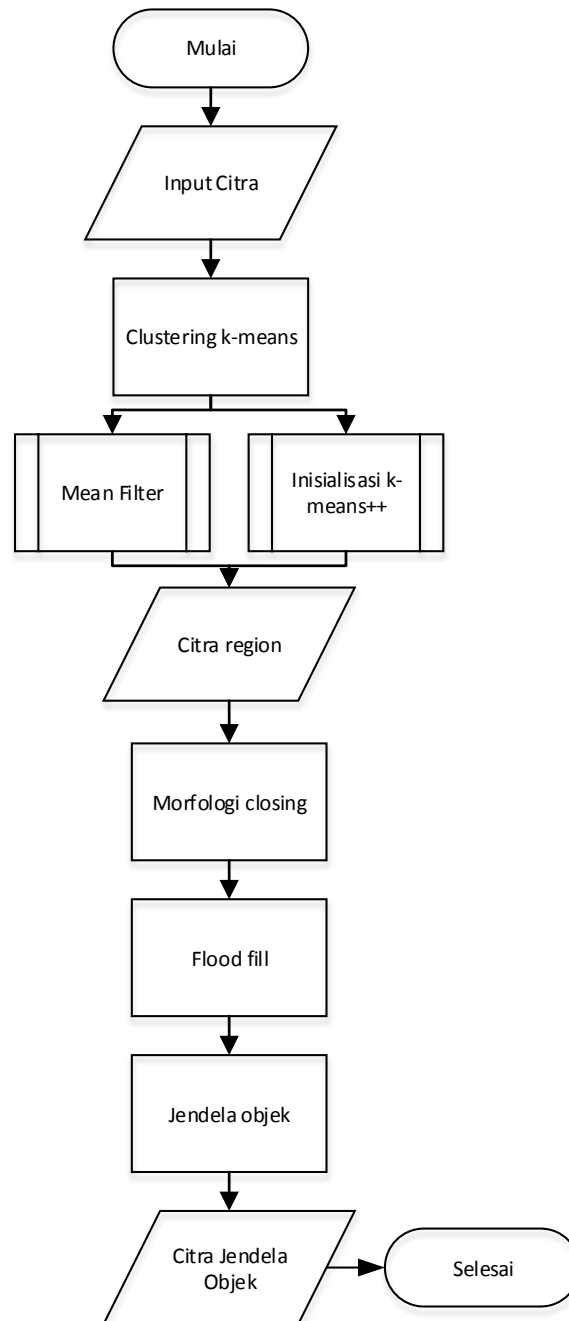
Pengambilan tekstur hanya pada objek bunga tentunya akan lebih fokus dalam mengenali jenis bunga tersebut karena tekstur yang diambil adalah dari objek bunga saja tidak keseluruhan piksel pada gambar yang bisa saja terdapat banyak sekali tekstur-tekstur beragam padahal objek jenis bunganya sama.

Segmentasi objek bunga ini dilakukan dengan menggunakan *k clustering* means. Dilakukan dengan hanya menggunakan dua *cluster*. Hipotesanya yaitu pada gambar dengan objek relatif terhadap background maka penggunaan dua *cluster* diharapkan dapat memisahkan objek dengan background. Dua *cluster* ini merupakan representasi dari *cluster* objek dan *cluster* background. Jadi piksel-piksel yang termasuk objek akan dikenali pada *cluster* objek begitu juga piksel-piksel yang termasuk background akan dikenali pada *cluster* background.

Tentunya pengambilan fitur akan sangat berperan penting dalam menghasilkan *cluster* yang baik. Oleh karena itu disini menggunakan pengambilan fitur warna dan tekstur. Dengan fitur warna adalah nilai a^* , b^* dari ruang warna Lab dan nilai H, S dan V dari ruang warna HSV. Sedangkan pemilihan tekstur menggunakan nilai-nilai dari GLCM.

Hasil segmentasi akan mendapatkan objek dimana untuk pengenalan jenis bunga tidak menggunakan objek hasil segmentasi ini. Hal ini dikarenakan hasil

segmentasi dapat tidak secara sempurna mendapatkan objek. Dikarenakan beberapa kondisi, seperti pencahayaan atau bayangan. Jadi segmentasi dijadikan sebagai pengukuran secara umum dimana bagian objek berada.



Gambar 3.7 Diagram alir tahapan segmentasi untuk mendapatkan jendela objek

Dengan mengetahui estimasi ini kemudian dilakukan pemberian jendela objek. Pemberian jendela objek mengatasi permasalahan segmentasi yang kurang bagus sekaligus memberikan sedikit ciri lingkungan pada objek bunga tersebut. Dikarenakan objek bunga yang sama memiliki kecenderungan lingkungan yang sama. Disnilah oleh sebab itu jendela objek ini yang dijadikan sebagai piksel-piksel dimana ekstraksi fitur dilakukan.

3.3.6. Ekstraksi Fitur

Setelah dilakukan tahapan *segmentasi* pada citra, didapatkan citra jendela objek. Pada bagian citra jendela objek kemudian dilakukan tahapan ekstraksi *fitur* yaitu *fitur* warna, *fitur* tekstur dan *fitur visual word*. *Fitur* warna yang diekstraksi adalah nilai a^* dan b^* dari ruang warna $L^*a^*b^*$ dan nilai warna H, S, dan V dari ruang warna *HSV*. Sedangkan *fitur* tekstur yang diekstraksi adalah nilai *gray level co-occurrence* yaitu nilai kontrasan, energi, *homogenitas* dan *korelasinya*.

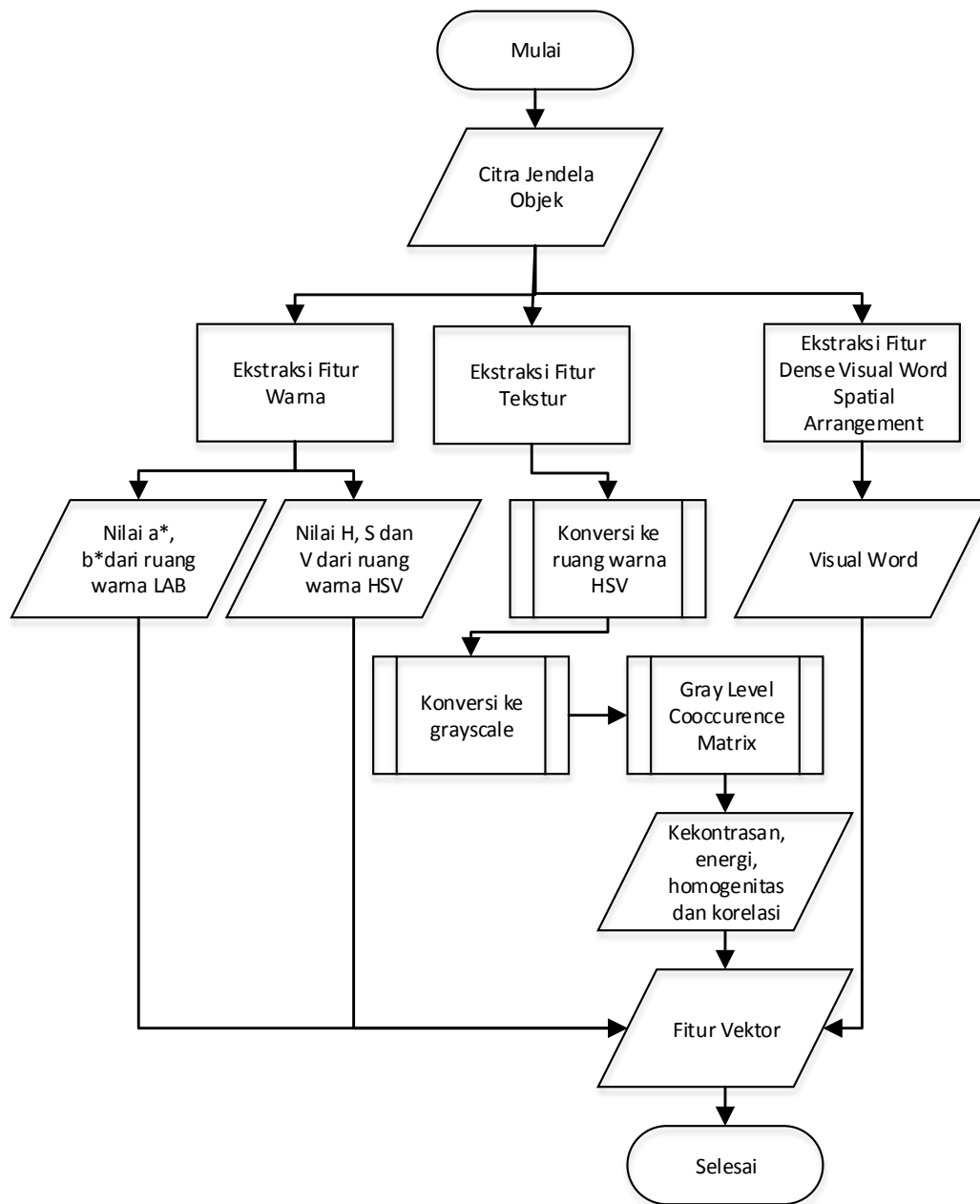
Diagram alir dari ekstraksi *fitur* dapat dilihat pada Gambar 3.8 dibawah. Pertama dimulai dengan input berupa citra jendela objek yang diekstraksi fitur warna, fitur tekstur dan ekstraksi fitur DVSA. Pada ekstraksi fitur warna hanya diambil nilai a^* dan b^* , dari ruang warna LAB dan nilai H,S,V dari ruang warna HSV. Pada balok ekstraksi fitur tekstur menandakan proses ekstraksi fitur tekstur yang inputnya adalah citra jendela objek.

Arah panah selanjutnya merupakan sub proses yang berada pada proses ekstraksi fitur tekstur. Jadi citra jendela objek tersebut dikonversi ke ruang warna HSV terlebih dahulu, kemudian setelah dikonversi ke ruang warna HSV hasil citra ruang warna baru ini dikonversi kembali menjadi citra abu-abu. Dari citra abu-abu ini diambil fitur GLCM menghasilkan data berupa fitur kontrasan, energi, homogenitas dan korelasi.

3.3.7. Evaluasi

Metode yang diajukan dievaluasi performanya dengan metode *bag of visual word* dan *visual word spatial arrangement* dengan pengujian 2, 4, dan 6 *words* pada

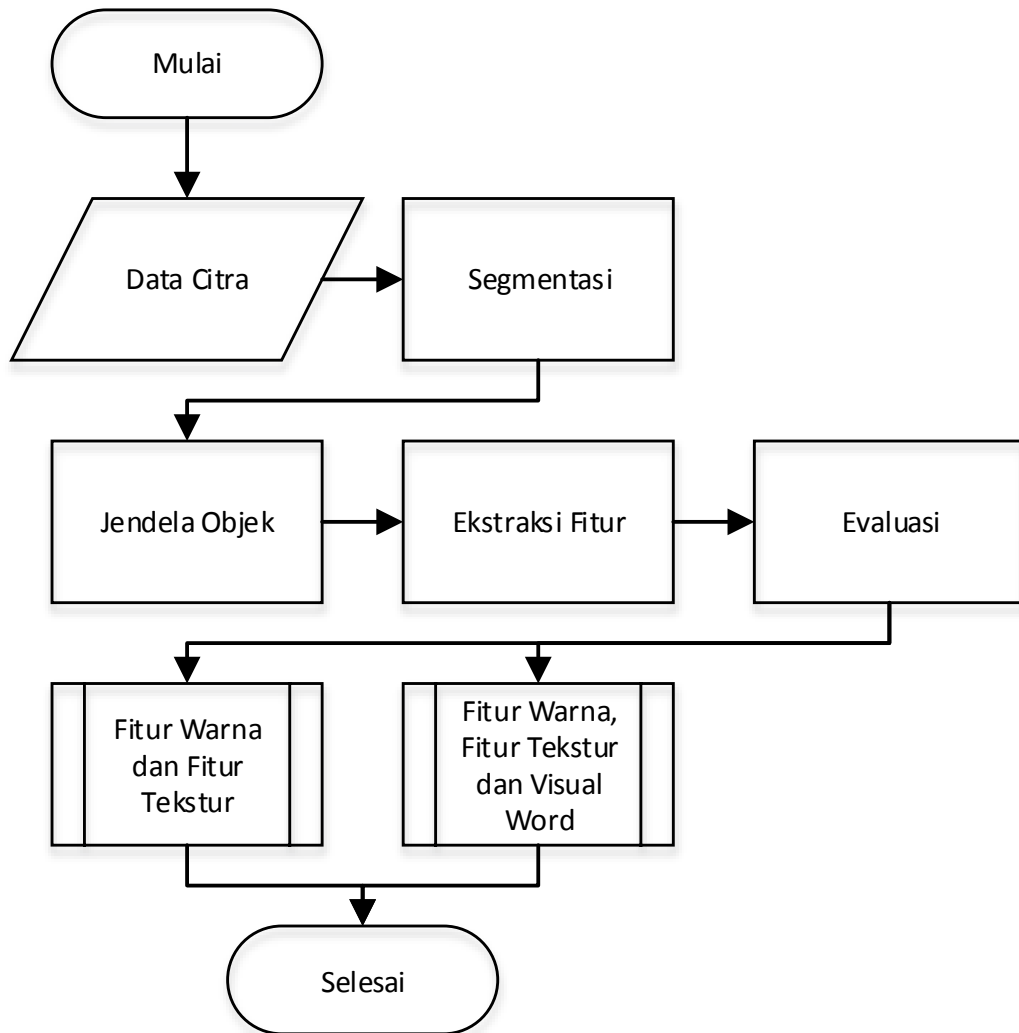
dataset 15-scenes. Pengklasifikasi menggunakan *Random Forest* dengan pembagian data *training* dan *testing* menggunakan *10-fold cross validation*



Gambar 3.8 Diagram tahapan ekstraksi fitur

Metode yang diajukan juga dievaluasi performanya bersama *fitur* warna dan tekstur pada dataset 30 jenis bunga. Diagram alir evaluasi ditunjukkan pada Gambar 3.9 dibawah ini. Pada Gambar 3.9 terdapat tiga diagram utama. Diagram ini menyatakan evaluasi. Pada diagram kiri atas, dimulai input berupa data citra

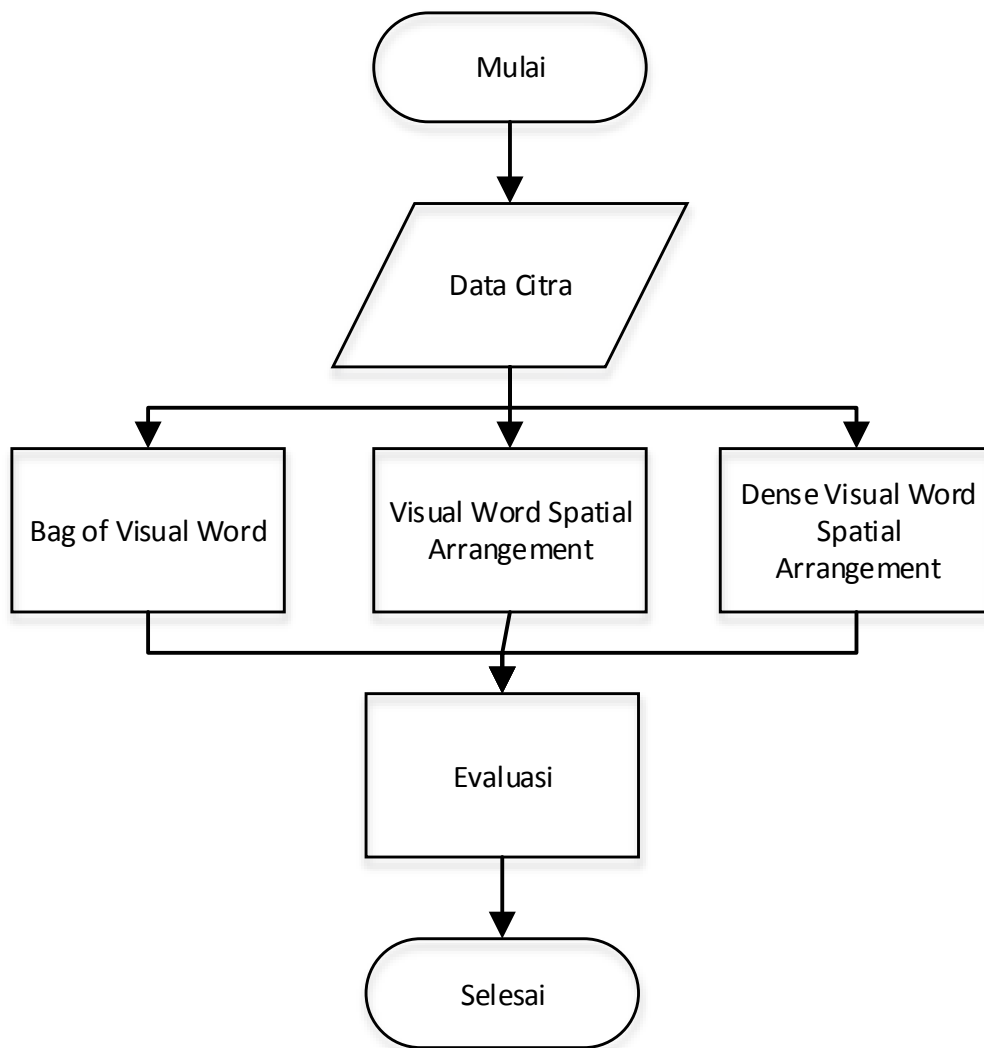
kemudian data citra ini dilakukan segmentasi. Hasil segmentasi kemudian diproses sehingga mendapat jendela objek. Ditunjukkan pada balok jendela objek. Setelah itu diproses ekstraksi fitur. Hasil dari ekstraksi fitur inilah yang berupa fitur vektor dimana dilakukan evaluasi. Ditunjukkan pada balok sub proses dari Evaluasi. Jadi yang dibandingkan adalah gabungan fitur warna dan tekstur dengan gabungan fitur warna, fitur tekstur dan visual *word*.



Gambar 3.9 Diagram tahapan evaluasi pertama

Untuk evaluasi pada citra umum ditunjukkan pada Gambar 3.10. Pertama dimulai dengan data citra sebagai inputan. Kemudian diproses dengan Bag of Visual *Word*, Visual *Word* Spatial Arrangement dan Dense Visual *Word* Spatial Arrangement. Dimana ditunjukkan pada balok setelah inputan data citra. Panah terakhir menunjuk ke balok evaluasi. Jadi hasil dari ketiga algoritma ini dievaluasi.

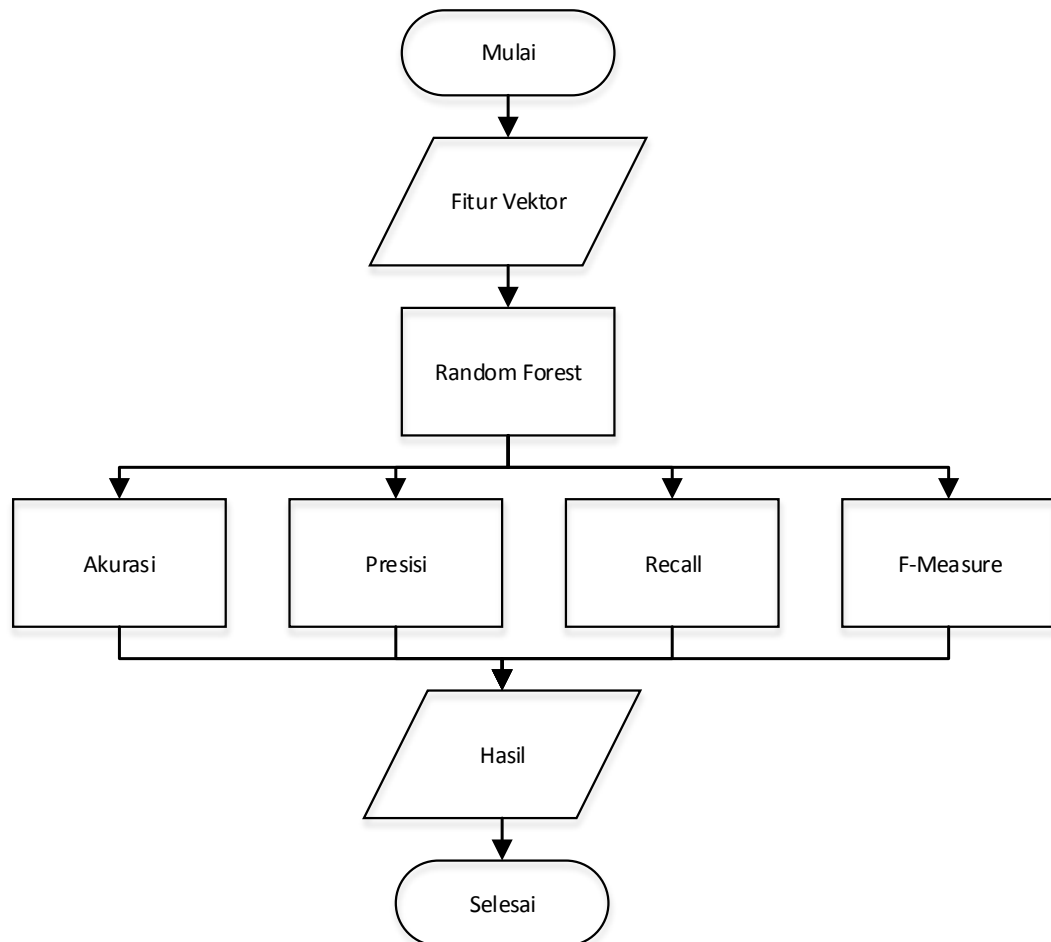
Pada Gambar 3.11 merupakan detail tahapan dari evaluasi. Dimulai inputan berupa fitur vektor . Fitur vektor ini merupakan hasil masing-masing algoritma pada proses sebelumnya. Dari fitur vektor ini kemudian diproses dengan algoritma klasifikasi.



Gambar 3.10 Diagram tahapan evaluasi pada citra umum

Klasifikasi yang digunakan adalah klasifikasi Random forest. Fitur vektor merupakan data yang ditunjukkan dengan bentuk trapesium sedangkan klasifikasi Random forest merupakan proses dimana ditunjukkan oleh bentuk balok pada diagram dari Gambar 3.11. Dari klasifikasi ini dihitung nilai evaluasi. Nilai evaluasi tersebut adalah nilai evaluasi akurasi, nilai evaluasi presisi, nilai evaluasi recall dan nilai evaluasi f-measure. Ditunjukkan dengan arah panah dari klasifikasi menuju ke

masing-masing balok proses dari evaluasi. Hasil berupa nilai evaluasi dimana ditunjukkan dengan arah panah dari masing-masing evaluasi akurasi, presisi, recall dan f-measure menuju ke arah balok data hasil.



Gambar 3.11 Diagram tahapan evaluasi secara detail

[Halaman ini sengaja dikosongkan]

BAB 4

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Pada bagian ini dipaparkan hasil implementasi dari setiap langkah yang telah dipaparkan pada Bab 3. Kemudian dilanjutkan dengan menunjukkan hasil dari uji coba. Setelah itu dipaparkan evaluasi dan pembahasan hasil yang diperoleh pada bagian akhir bab ini.

4.1.1. Membuat Dictionary

Program utama untuk membuat dictionary dijelaskan sebagai berikut. Terdapat perbedaan dalam membuat dictionary untuk metode yang diusulkan dengan metode BoV dan WSA. Pada bagian program membuat *Dictionary*, pertama dibuat *variabel testPath*. Dimana *variabel* ini menunjukkan folder tempat gambar-gambar untuk *Dictionary* dibuat. *Testfilename* merupakan nama file beserta direktorinya. Setelah gambar dibaca dengan *imread* maka gambar tersebut diperkecil dengan ukuran setengah dari gambar asli.

Dapat dilihat pada kode program yaitu pada baris *im=imresize*. Kemudian dilakukan perhitungan *deskriptor lokal* pada keseluruhan gambar. Ditunjukkan pada fungsi *vl_sift* dan *vl_sift*. Fungsi *vl_sift* digunakan ketika membuat *Dictionary* untuk BoVW dan WSA sedangkan fungsi *vl_sift* digunakan ketika membuat *Dictionary* untuk metode yang diusulkan yaitu DVSA. Jadi pada program dilakukan *uncomment* ke salah satu baris ini jika baris tersebut tidak digunakan.

Setelah dilakukan perulangan ke semua gambar maka kode *break* ditulis untuk keluar dari perulangan *while* sehingga setelah itu dapat dilakukan operasi pemilihan *Dictionary*. Pada baris *acak = randperm* dilakukan pemilihan *Dictionary* secara acak. Jadi baris ini adalah inisialisasi jumlah, yang digunakan kemudian pada kode di bawahnya yaitu perulangan *for i=1:word*. Perulangan ini menyatakan pemilihan *Dictionary* dari banyak deskriptor lokal yang sebelumnya telah dihasilkan. Ditunjukkan pada kode program *Dict(I,:)=desc(acak(i),:)*. Baris

terakhir yaitu *toc(startTime)* digunakan untuk mencatat waktu dari awal program sampai selesai. Lebih lanjut dapat dilihat pada keterangan pada program.

```
%Oleh gama wisnu fajariantanto
% Membuat dictionary
function [dict,desc]=encodingDict(word)

testPath = 'building/';
testImageList = dir(testPath);
if isempty(testImageList)
error('direktori tidak ada.');
```



```
end

imageCount = 0;
imageIndex=1;
desc = zeros(0);
startTime = tic;
while 1
if testImageList(imageIndex).name(1)=='.'
imageIndex=imageIndex+1;
continue;
end
testFilename = [testPath testImageList(imageIndex).name];
    %skala ukuran
clear im;
    %im=imread(testFilename);
im=imresize(imread(testFilename),0.5);
    %membuat dictionary untuk BoV dan WSA
    [f, d] = vl_sift(single(im)) ;
    %membuat dictionary untuk metode yang diusulkan (DVSA)
    %[f, d] = vl_dsift(single(im),'size',10,'step',20) ;
if imageCount>0
desc = [desc d];
else
desc = d;
end

clear fitur
imageIndex=imageIndex+1;
imageCount = imageCount+1;
if imageCount > jumlahImage
```

```

break;
end
end
desc=desc';
acak = randperm(size(desc,1));
for i=1:word
dict(i,:)=desc(acak(i),:);
end
toc(startTime)
end

```

4.1.2. Mendapatkan *Word* dan Posisinya

Pada program *encodingExtractOri* yang ditunjukkan pada program dibawah merupakan kode untuk mendapatkan *word* beserta posisi dari *word* tersebut. Jadi hasilnya disimpan pada *matrixwords* dan *posit*. Sebagaimana ditunjukkan pada baris awal fungsi yaitu *function [words, posit] = encodingExtractOri (Dict, folder,nCitra)*. *Dict*, *folder* dan *nCitra* masing-masing menunjukkan fungsi ini butuh *inputDict, folder*, dan *nCitra*.

Dict adalah *matrixDictionary*, *folder* merupakan *string* alamat folder setiap kelas dari gambar yang akan dijadikan data *training*. Sedangkan *nCitra* merupakan jumlah gambar pada folder tersebut. Kode pada program ini bagian awal-awal mirip dengan kode sebelumnya yaitu sampai membentuk deskriptor dari gambar. Jadi ada variabel *testfilename* , *imread*, dan *imresize* pada awal kode.

Setelah itu ada *vl_sift* yang digunakan untuk mendapatkan deskriptor pada gambar. Parameter dari *vl_sift* yang digunakan adalah *single(im)*, yaitu mengubah double pada *im* menjadi tipe *single*. Kemudian terdapat *PeakThresh* dengan nilai 10 yang digunakan untuk perhitungan *interest point*. Semakin tinggi nilai maka *interest point* yang digunakan semakin sedikit dan hasil detektor *interest point*nya seperti hanya yang semakin terbaik saja. Fungsi *try catch* digunakan untuk membedakan pengolahan gambar yang berwarna dan gambar yang *grayscale*. Kemudian baris *[drops, words]=min(vl_alldist2)* digunakan untuk menghitung jarak antara deskriptor dan *Dictionary*. Setelah itu if else *imageCount* digunakan

membedakan pengolahan gambar pertama dan gambar selanjutnya untuk pembuatan *matrix* yang akan digunakan nantinya sebagai fitur vektor.

Variabel dibawahnya yang mengikuti, yaitu variabel, *posit*, *word* dan *words* digunakan sebagai penempatan indeks *variabel* untuk posisi dan *word*. Sebagaimana program sebelumnya break ditambahkan ketika gambar dalam folder tersebut sudah diproses semua. Bagian terakhir `toc(startTime)` digunakan untuk mencatat waktu.

```
%Oleh gama wisnu fajariantanto
%Ekstraksi Fitur
function [words,posit]=encodingExtractOri(dict, folder,nCitra)

testPath = folder;
testImageList = dir(testPath);
if isempty(testImageList)
error('direktori tidak ada.');
```

end

```
imageCount = 0;
imageIndex=1;
posit = zeros(0);
startTime = tic;
words = zeros(0);
%-----
while 1
if testImageList(imageIndex).name(1)=='.'
imageIndex=imageIndex+1;
continue;
end
testFilename = [testPath testImageList(imageIndex).name];
    %skala ukuran
clear im;
    %im=imread(testFilename);
im=imresize(imread(testFilename),0.5);
try
    %[position, descriptor] = vl_dsift(single(im),'size',10,'step',20) ;
    [position, descriptor] = vl_sift(single(im),'PeakThresh', 10) ;
catch
    [position, descriptor] = vl_sift(single(rgb2gray(im)),'PeakThresh', 10) ;
```

```

%[position, descriptor] =
vl_dsift(single(rgb2gray(im)), 'size', 10, 'step', 20) ;
end

[drops, word]=min(vl_alldist2(descriptor, dict'), [], 1);
if imageCount>0
position(5,:)=imageCount+1;
posit = [posit position];
word(2,:)=imageCount+1;
words = [words word];
else
position(5,:)=imageCount+1;
posit = position;
word(2,:)=imageCount+1;
words = word;
end

clear drops
imageIndex=imageIndex+1;
imageCount = imageCount+1;
if imageCount > nCitra-1
posit(3:4,:)=[];
break;
end
end
toc(startTime)
end

```

4.1.3. Bag of Visual Word

Pada fungsi *fiturBag=encodingBag(words,nword)* memiliki *inputwords* dan *outputvariabel* *fiturBag*. Bag of Visual Word ini memiliki dua fungsi yaitu *encodingBag* dan *fiturBagAll*. *FiturBag* merupakan fungsi untuk membuat fitur vektor akhir dari BoVW. Terdapat dua perulangan bercabang yaitu *for k=1:numImage* dan *for i=1:nword*.

Perulangan pertama untuk memproses semua gambar dan perulangan kedua untuk memproses setiap *word* dari gambar tersebut. *fiturBag(k,i)=sum(words_(words_(:,2)==k))* digunakan untuk menjumlah *word* dari gambar yang sama. Pada fungsi *fiturBagAll(jenis,word)* terdapat *variabel*

prepareArray. Variabel ini digunakan untuk me *load* variabel encoding yang sebelumnya telah dihasilkan. Setelah dilakukan *encodingBag* maka variabel prepareArray digunakan untuk menyimpan nama dari hasil fiturBag ini. Tertulis pada baris selanjutnya yaitu *save(prepareName, 'fiturBag')*.

```
%Oleh gama wisnu fajariantanto
%Membuat bag of word encoding dari word yang dihasilkan sebelumnya
%(encodingExtractOri)
function [fiturBag]=encodingBag(words,nword)

words_=words';
numImage=max(words_(:,2));
for k=1:numImage
for i=1:nword
fiturBag(k,i)=sum(words_(words_(:,2)==k,1)==i);
end
end

function fiturBagAll(jenis,word)
for i=1:jenis
startTime = tic;
prepareArray = ['dict_2_' num2str(i) '_enc_ext_sift_image' ];
load(prepareArray);
[fiturBag]=encodingBag(words,word);
prepareName = ['fiturBag' num2str(i)];
save(prepareName,'fiturBag');
clear words fiturBag
end
```

4.1.4. Visual Word Spatial Arrangement

Pada fungsi *fiturSpatialAll(nword, jenis)* terdapat dua parameter *input* yaitu *nword* dan *jenis*. *Nword* adalah *input* untuk berapa *word* yang digunakan sedangkan *jenis* adalah *input* untuk berapa jenis atau kelas label yang digunakan pada data *training*. *fiturSpatialAll* me *load* hasil perhitungan deskriptor dari kode sebelumnya kemudian memanggil fungsi *encodingSpatial* untuk mendapatkan informasi *spatial*. Setelah itu hasilnya disimpan, yaitu pada baris *save(prepareName, 'wordsarrange')*.

Pada fungsi *constfiturSpatialAlls* memiliki hasil *output* *fiturVectorSpatial* dari visual *wordspatial* arrangement. Input parameter adalah jenis. Sama seperti *fiturSpatialAll* jenis adalah *inputan* untuk berapa jenis atau kelas label yang digunakan pada data *training*. Fungsi *constfiturSpatialAlls* ini me *load* hasil dari visual *wordspatial* arrangement dari kode sebelumnya kemudian dibentuk fitur vektor. Jadi fungsi *constfiturSpatialAlls* ini membentuk fitur vektor yang akan digunakan untuk proses klasifikasi.

```
function fiturSpatialAll(nword,jenis)
for i=1:jenis
prepareArray = ['dict_2_' num2str(i) '_enc_ext_sift_image' ];
load(prepareArray);
[wordsarrange]=encodingSpatial(posit,words,nword);
prepareName = ['fiturspatialDuawo' num2str(i)];
save(prepareName,'wordsarrange');
clear posit words wordsarrange
end

function fiturVectorSpatial= constfiturSpatialAlls(jenis)
for i=1:jenis
prepareArray = ['fiturspatialDuawo' num2str(i)];
load(prepareArray)
w1(:,1:4)=wordsarrange(:,1,:);
w2(:,1:4)=wordsarrange(:,2,:);
fiturSpatial=[w1 w2];
fiturSpatial(:,end+1)=i;
fitur = fiturSpatial;
if i>1
fiturVectorSpatial = [fiturVectorSpatial;fitur];
else
fiturVectorSpatial = [fitur];
end
clear w1 w2 fiturSpatial
end
```

4.1.5. Dense Visual Word Spatial Arrangement

Pada *dense visual word spatial arrangement* ini terdapat dua fungsi yaitu *fiturDenseSpatialAll* dan *fiturVectorSpatial*. Sama seperti pada sebelumnya *fiturVectorSpatial* membentuk fitur vektor yang akan digunakan untuk proses

klasifikasi. Dengan me *load* terlebih dahulu hasil dari informasi *spatial* pada fungsi *fiturDenseSpatialAll*.

Pertama dilakukan perulangan sebanyak jumlah kelas label yang terdapat pada data *training* yaitu dapat dilihat pada baris awal *fiturDenseSpatialAll* for *i=1:jenis*. Di perulangan ini me *load* deskriptor yang dihasilkan oleh proses sebelumnya kemudian dilakukan pengambilan informasi *spatial* yang ditulis dengan *[wordsarrange]=encodingSpatial(posit,words,nword)*. Hasilnya disimpan dengan menuliskan *save(prepareName,'wordsarrange')*. Jadi baris ini menyimpan variabel *wordsarrange* dengan nama *prepareName*. *PrepareName* memiliki penamaan identik dengan jenis kelas label.

```
function fiturDenseSpatialAll(nword, jenis)
for i=1:jenis
startTime = tic;
prepareArray = ['dict_2_' num2str(i) '_enc_ext_DENSift_image' ];
load(prepareArray);
[wordsarrange]=encodingSpatial(posit,words,nword);
prepareName = ['fiturDENspatialDuawo' num2str(i)];
save(prepareName,'wordsarrange');
clear posit words wordsarrange
end
```

```
function fiturVectorSpatial= constfiturProp1Alls(jenis)
% startTime = tic;
for i=1:jenis
prepareArray = ['fiturDENspatialDuawo' num2str(i)];
load(prepareArray)
w1(:,1:4)=wordsarrange(:,1,:);
w2(:,1:4)=wordsarrange(:,2,:);
fiturSpatial=[w1 w2];
fiturSpatial(:,end+1)=i;
fitur = fiturSpatial;
if i>1
fiturVectorSpatial = [fiturVectorSpatial;fitur];
else
fiturVectorSpatial = [fitur];
end
clear w1 w2 fiturSpatial
```


end

4.2 Uji Coba dan Hasil

Bag of Visual *Word*, Visual *Word* Spatial Arrangement dan metode yang diusulkan diimplementasikan dengan menggunakan Matlab versi 2012b. Juga menggunakan fungsi API dari VLFeat versi 0.9.17. Untuk evaluasi pengujian menggunakan Weka 3.6. Dibangun di atas platform Windows 8.1, dengan spesifikasi processor Intel® Core™ 2 Duo CPU T6670 2.20 GHz dan Memory 4 GB. Pada bagian selanjutnya dari sub bab ini akan dijelaskan analisa uji coba dan hasilnya secara keseluruhan. Kemudian dijelaskan per sub bagian hasil tiap uji coba.

4.2.1. Analisa Uji Coba dan Hasil Secara Keseluruhan Pada Skenario Pertama

Skenario pertama dilakukan dengan tujuan untuk mengetahui apakah metode yang diusulkan lebih baik dari metode dasar dan metode *spatial* lainnya. Apakah metode yang diusulkan kompetitif dibanding metode *word* lain. Oleh karena itu metode yang diusulkan dievaluasi dengan metode dasar yaitu metode yang menjumlahkan *word* dan metode *spatial*. Kedua metode ini merupakan representasi metode *word* yang ada.

Tujuan selanjutnya adalah apakah dengan hanya menggunakan sedikit *word* dengan kata lain sedikit fitur vektor akan dapat mengenali jenis gambar. Dilihat dengan hasil nilai evaluasi performa. Dimana dengan fitur vektor yang sedikit berdampak pada cepatnya perhitungan. Selain itu digunakan Random Forest sebagai *klasifier* yang kompetitif. Dengan 10 *fold cross validation* pada 4485 citra umum dimana 10 *fold cross validation* ini diharapkan dapat memberikan gambaran evaluasi yang lebih menyeluruh. Karena membagi data menjadi 10 bagian dengan ada yang bagian sebagai *training* dan sisanya testing. Dilakukan secara berulang pada keseluruhan data, dengan bagian yang diambil sebagai *training* tidak diambil lagi menjadi *training* begitu juga dengan testingnya.

Penjelasan kedua alasan tersebut tergambar pada skenario uji coba pertama yaitu penggunaan *word* berjumlah 2, 4 dan 6 serta pembandingan metode dengan Bag of visual *word* dan *Visualwordspatial* arrangement.

Hasil uji coba untuk skenario pertama yaitu metode yang diusulkan unggul pada semua uji coba terhadap metode dasar. Metode *spatial* lain juga unggul pada semua uji coba terhadap metode dasar. Unggul baik dalam hal akurasi, presisi, recall maupun f-measures. Peningkatan performa akurasi diperoleh sampai 36.20% dari metode yang diusulkan terhadap metode dasar. Dengan metode dasar 2 *word* dan metode yang diusulkan 6 *word*.

Untuk jumlah *word* yang sama metode yang diusulkan terdapat peningkatan performa akurasi sebesar 30.99% terhadap metode dasar dengan sama-sama menggunakan 4 *word*. Metode *spatial* lain yaitu WSA unggul terhadap semua metode pada jumlah *word* yaitu 2 *word*. Selain itu metode yang diusulkan unggul baik terhadap metode dasar maupun metode WSA. Didapat peningkatan akurasi sebesar 18.75% dari metode yang diusulkan terhadap metode WSA. Peningkatan ini ditemui pada semua evaluasi yaitu akurasi, presisi, recall dan f-measures.

4.2.2. Analisa Uji Coba dan Hasil Secara Keseluruhan Pada Skenario Kedua

Skenario kedua dilakukan dilakukan dengan tujuan untuk mengetahui apakah fitur dari *word* dapat digabungkan dengan fitur dari metode ekstraksi fitur lain. Dilihat dengan bagaimana performa yang dihasilkan dari penggabungan tersebut. Apakah lebih baik dari metode ekstraksi fitur yang sudah ada dan juga bagaimana performa antar metode *word* yang berbeda. Apakah fitur *word* dapat digabungkan dengan fitur ekstraksi pada umumnya. Dengan tidak memberikan beban fitur vektor yang besar pada fitur umum yang dihasilkan oleh ekstraksi fitur warna dan tekstur.

Sama seperti penjelasan pada skenario pertama disini juga menggunakan klasifier yang sama dan evaluasi yang sama. Yaitu pengklasifikasi Random Forest

dan evaluasi 10 *fold cross validation*. Kemudian ekstraksi fitur yang digunakan adalah ekstraksi fitur warna dan ekstraksi fitur bentuk.

Metode ekstraksi fitur yang digunakan adalah ekstraksi fitur warna dan fitur tekstur sebagai metode dasar dan dibandingkan dengan metode ekstraksi fitur warna, tekstur dan *word* sebagai metode gabungan. Dengan dua *word* dari metode *bag of visual word*, *visual words spatial arrangement* dan metode yang diajukan.

Pada skenario yang kedua, akurasi secara berturut-turut untuk penggabungan fitur dengan dua *word* metode dasar BoVW, penggabungan dengan WSA, penggabungan dengan DVSA dan tanpa penggabungan diperoleh akurasi sebesar 81.67%, 85.50%, 82.33% dan 82.00%. Presisi sebesar 80.90%, 85.30%, 81.20% dan 81.10%. Recall sebesar 81.70%, 85.50%, 82.30% dan 82.00%. F-Measures sebesar 81.00 %, 85.00%, 81.60% dan 81.20%.

Hasil menunjukkan penggabungan ekstraksi fitur dengan WSA menunjukkan performa terbaik yaitu secara berturut-turut untuk akurasi, presisi, recall dan f-measure yaitu 85.50%, 85.30%, 85.50% dan 85.00%.

Urutan terbaik kedua diperoleh oleh penggabungan ekstraksi fitur dengan metode yang diusulkan yaitu secara berturut-turut nilai akurasi, presisi, recall dan f-measurenya adalah 82.33%, 81.20%, 82.30% dan 81.60%. Sedangkan penggabungan ekstraksi fitur dengan metode BoVW menghasilkan secara berturut-turut nilai akurasi, presisi, recall dan f-measure sebesar 81.67%, 80.90%, 81.70% dan 81.00%.

Pada skenario kedua ini ternyata penggabungan dengan BoVW tidak dapat meningkatkan akurasi terhadap metode dasar. Sedangkan penggabungan menggunakan metode *spatial* maupun metode yang diajukan dapat meningkatkan performa dari metode dasar. Pada semua uji coba penggunaan metode *word* yang spasial dapat meningkatkan performa terhadap metode dasar. Sedangkan tidak semua metode *word* dapat meningkatkan performa terhadap metode dasar.

Penggabungan metode *word* dengan fitur umum dimana metode *word* yang hanya melakukan penjumlahan membuat performa menurun dari metode dasar tanpa penambahan *word*. Dapat diberikan penjelasan pada saat menggunakan *spatial* artinya ciri ruang pada gambarlah yang di kode kan sehingga menjadi ciri pada gambar tersebut. Metode dasar yang hanya menggunakan jumlahan *word* dapat memberikan seperti *noise* apabila digabungkan dengan metode ekstraksi fitur maupun ketika digunakan sendiri tanpa digabungkan dengan fitur lain. Sebagai contoh apabila menggunakan dua *word*. Maka pada gambar tertentu akan memberikan ciri hanya jumlah *word* pertama dan jumlah *word* kedua, yang dapat terlihat seperti angka *noise* atau random.

WSA dan DVSA merupakan metode *word* yang *spatial*. Kedua metode menghasilkan jumlah *word* yang sama yaitu 4 dimana adalah mewakili kuadran dikali banyaknya *word* yang digunakan. Dengan kata lain jumlah fitur vektor yang dihasilkan adalah $4W$ dengan W adalah jumlah *word*. *Scaling* akan membuat gambar tampak lebih besar atau lebih kecil dari skala yang sebenarnya sedangkan rotasi memberikan bentuk gambar yang terotasi. Sesuai dengan skala rotasi yang diberikan. Ditemui dengan hasil evaluasi yang dihasilkan, model ditemui kesulitan ketika gambar dirotasi. Ditemui kesalahan dalam mengenali gambar yang dirotasi. Dengan gambar yang tidak dirotasi sebelumnya benar ketika dikenali.

Metode *spatial* lain yang menjadi pembanding atau WSA, pada semua skenario selalu unggul saat *word* berjumlah dua. Dengan kata lain ketika hanya sedikit sekali *word* yang digunakan maka metode tersebut unggul terhadap semua metode untuk semua skenario. Dapat diberikan penjelasan dikarenakan metode WSA menggunakan *interest point* yang dapat fokus sehingga titik-titiknya akan banyak yang saling berdekatan. Jadi ketika menggunakan dua *word* akan menjadi sederhana dan lebih mengelompokkan anggota mana yang menjadi *word* pertama dan mana yang menjadi *word* kedua, dengan menggunakan *interest point* ini. Berbeda ketika seperti metode yang diajukan dimana titik-titik akan lebih melebar luas seperti merata pada keseluruhan gambar. Jadi dua *word* yang memiliki

pembeda ciri yang sangat sedikit sekali ini ketika diberikan secara merata pada titik-titiknya maka akan kurang dapat membedakan ciri dari gambar tersebut dibandingkan dengan titik-titik yang mengelompok.

4.2.3. Hasil Metode yang diusulkan terhadap BoVW dan WSA

Detail hasil Metode yang diusulkan terhadap BoVW dan WSA dapat dilihat pada Tabel 4.1 di bawah

Tabel 4.1 Hasil Evaluasi Metode BoVW, WSA dan DVSA

Jumlah	Metode BoVW			
Word	Akurasi	Presisi	Recall	F-Measure
2	30.82 %	22.10 %	30.80 %	23.20 %
4	36.89 %	35.20 %	36.90 %	35.60 %
6	37.03 %	34.00 %	37.00 %	34.90 %
Jumlah	Metode WSA			
Word	Akurasi	Presisi	Recall	F-Measure
2	46.45 %	44.00 %	46.50 %	44.10 %
4	49.72 %	47.80 %	49.70 %	47.20 %
6	48.27 %	46.60 %	48.30 %	45.60 %
Jumlah	Metode yang diusulkan (DVSA)			
Word	Akurasi	Presisi	Recall	F-Measure
2	43.51 %	41.10 %	43.50 %	41.40 %
4	67.88 %	66.90 %	67.90 %	66.90 %
6	67.02 %	66.30 %	67.00 %	66.00 %

Dapat dilihat pada Tabel 4.1 metode DVSA unggul dalam semua *word* yang diuji terhadap metode dasar BoVW. Sedangkan WSA unggul terhadap DVSA ketika menggunakan *word* yang sangat sedikit yaitu 2 *word*.

4.2.4. Hasil Peningkatan Performa pada DVSA, BoVW dan WSA

Detail hasil Metode yang diusulkan terhadap BoVW dan WSA dapat dilihat pada Tabel 4.2 di bawah

Tabel 4.2 Hasil Peningkatan Metode untuk Setiap *Word*

Metode	Jumlah Word	Peningkatan Performa Terhadap BoVW			
		Akurasi	Presisi	Recall	F-Measure
WSA	2	15.62 %	21,90 %	15,70 %	20,90 %
WSA	4	12.82 %	12,60 %	12,80 %	11,60 %
WSA	6	11.23 %	12,60 %	11,30 %	10,70 %
DVSA	2	12.68 %	19,00%	12,70%	18,20 %

DVSA	4	30.99 %	31,70 %	31,00 %	31,30 %
DVSA	6	29.98 %	32,30 %	30,00 %	31,10 %
Metode	Jumlah	Peningkatan Performa Terhadap DVSA			
	<i>Word</i>	Akurasi	Presisi	Recall	F-Measure
WSA	2	2.94 %	2.90 %	3.00 %	2.70 %
Metode	Jumlah	Peningkatan Performa Terhadap WSA			
	<i>Word</i>	Akurasi	Presisi	Recall	F-Measure
DVSA	4	18.16 %	19.10 %	18.20 %	19.70 %
DVSA	6	18.75 %	19.70 %	18.70 %	20.40 %

Dapat dilihat pada Tabel 4.2 peningkatan performa akurasi diperoleh oleh DVSA terhadap BoVW meningkat sampai 30.99 %. Sedangkan peningkatan performa DVSA terhadap WSA sampai 18.75%.

4.2.5. Hasil Performa Penerapan DVSA, BoVW dan DVSA bersama Fitur Warna dan Tekstur pada Pengenalan Objek Bunga

Hasil *confusion matrix* tanpa tambahan fitur *word* dapat dilihat pada Gambar 4.1. Untuk hasil *confusion matrix* dengan tambahan fitur *word* dari wsa dapat dilihat pada Gambar 4.2. Sedangkan hasil *confusion matrix* dengan tambahan fitur *word* dari DVSA dapat dilihat pada Gambar 4.3. Dan hasil *confusion matrix* dengan tambahan fitur *word* dari BoVW dapat dilihat pada Gambar 4.4. Dapat dilihat pada Gambar 4.1 bagian kolom atas adalah a, b, c, e sampai ad dimana merupakan singkatan dari penamaan label jenis bunga. Ditunjukkan pada sisi kanan untuk setiap baris. Secara berurutan adalah bunga bawang sebrang, bunga ceplok piring, bunga delapan dewa sampai dengan yang terakhir adalah bunga kamboja dan bunga kertas.

Terlihat bagian diagonal adalah bagian dengan angka-angka yang besar. Menunjukkan secara visual model dapat mengklasifikasi lebih banyak benar daripada yang salah. Bagian yang salah adalah bagian angka yang bukan terletak pada diagonal. *Word* yang digunakan adalah 2 *word*. Dari hasil yang ditunjukkan pada gambar 4.1, 4.2, 4.3 dan 4.4 dengan kesimpulan akurasi pada Tabel 4.3. Peningkatan performa oleh DVSA didapat sebesar 0.33% terhadap metode tanpa menggunakan *word*. Sedangkan penggunaan penambahan fitur WSA pada fitur warna dan tekstur dapat meningkatkan performa akurasi sebesar 3.50%. Untuk

penggunaan penambahan fitur BoVW akurasi turun sebesar 0.33%. Pola hasil ini sama seperti halnya pada saat hanya menggunakan fitur *word* sebagaimana ditunjukkan pada Tabel 4.2 yaitu ketika hanya menggunakan 2 *word* WSA unggul terhadap DVSA dan BoVW. Sedangkan DVSA hanya unggul terhadap BoVW

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad		
28	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	a = bawang sebrang	
5	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = ceplok piring	
0	0	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	c = delapan dewa	
0	0	4	2	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	d = chamaecrista fasciculata	
1	0	0	1	19	3	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	e = bunga sepatu putih	
0	0	0	0	2	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	f = bunga batavia	
0	0	0	0	0	4	16	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	g = bugenvil merah muda	
0	0	0	0	0	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	h = bunga kamboja
0	0	0	0	0	0	0	0	2	1	1	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	i = bugenvil putih
0	0	0	0	1	0	0	0	0	5	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	j = bunga kupu-kupu
0	0	0	0	0	0	0	0	0	0	15	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	k = Bunga sepatu merah
0	0	0	0	0	0	0	0	0	0	2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	l = Rolia pink
0	0	0	0	1	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = Bunga rumput
0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	n = melati
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = soka kuning
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	22	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	p = soka merah
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	19	0	0	2	0	0	0	0	0	0	0	0	0	0	1	q = dadap merah
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	r = bunga mentega
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	22	0	2	0	0	0	0	0	0	0	0	0	0	0	s = bunga sepatu kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	19	0	0	1	0	0	0	0	0	0	0	0	0	0	t = bunga merak
0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	u = widelia biflora
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	17	0	0	0	0	0	1	0	0	0	0	0	v = bunga iris kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	w = lili
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	x = bunga tapak dara
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	y = bunga putri malu
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	z = pseuderanthemum reticulatum
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	1	0	0	0	aa = bunga tomat	
2	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	13	0	0	0	0	ab = bunga biduran	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	ac = bunga kamboja jepang	
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	20	0	ad = bunga kertas

Gambar 4.1 Hasil confusion matrix tanpa tambahan fitur *word*

Dapat dilihat pada Gambar 4.2, sebagaimana dijelaskan pada keterangan sebelumnya untuk Gambar 4.1, pada bagian kolom atas adalah a, b, c, e sampai ad dimana merupakan singkatan dari penamaan label jenis bunga. Ditunjukkan pada sisi kanan untuk setiap baris. Secara berurutan adalah bunga bawang sebrang, bunga ceplok piring, bunga delapan dewa sampai dengan yang terakhir adalah bunga kamboja dan bunga kertas. Terlihat bagian diagonal adalah bagian dengan angka-angka yang besar. Menunjukkan secara visual model dapat mengklasifikasi lebih banyak benar daripada yang salah. Bagian yang salah adalah bagian angka yang bukan terletak pada diagonal. Berbeda dengan Gambar 4.1 pada Gambar 4.2 lebih banyak mengenali jenis dengan benar. Sebagai contoh untuk 3 bunga yang pertama yaitu bunga bawang sebrang, bunga ceplok piring, dan bunga delapan dewa pada Gambar 4.1 mengklasifikasikan dengan benar secara berurutan sebesar 28, 10, 8. Sedangkan pada Gambar 4.2 dapat mengklasifikasikan secara benar yaitu sejumlah 31,10 dan 9. Untuk seterusnya dapat dilihat pada Gambar 4.2 dibawah.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad		<-- classified as		
31	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0		a = bawang sebrang		
4	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0		b = ceplok piring	
1	0	9	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		c = delapan dewa	
0	0	1	5	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		d = chamaecrista fasciculata	
0	0	0	0	1	19	3	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0		e = bunga sepatu putih	
0	0	0	0	0	13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		f = bunga batavia	
0	0	0	0	0	3	19	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		g = bugenvil merah muda	
0	0	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		h = bunga kamboja	
0	0	0	0	0	0	1	1	2	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0		i = bugenvil putih
0	0	0	0	1	0	0	0	0	6	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	2	0	0	0	0	0	0		j = bunga kupu-kupu
0	0	0	0	1	0	0	0	1	0	0	15	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0		k = Bunga sepatu merah
0	0	0	0	0	0	0	0	0	0	0	2	18	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		l = Rolia pink
0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		m = Bunga rumput
0	0	1	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0		n = melati	
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		o = soka kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		p = soka merah
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		q = dadap merah
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0		r = bunga mentega
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	2	0	0	0	0	0	0	0	0	0	0	0		s = bunga sepatu kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	19	0	1	0	0	0	0	0	0	0	0	0	0		t = bunga merak
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0		u = widelia biflora
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	1	0	0	0	0	0		v = bunga iris kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0		w = lili
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0		x = bunga tapak dara
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0		y = bunga putri malu
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0		z = pseudanthemum reticulatum
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14	1	0	0	0		aa = bunga tomat	
3	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	1	13	0	0	0		ab = ceplok piring8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0		ac = bunga kamboja jepang
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	20	0		ad = bunga kertas

Gambar 4.2 Hasil confusion matrix dengan tambahan fitur worddari WSA

Pada Gambar 4.3, sebagaimana dijelaskan pada keterangan sebelumnya untuk Gambar 4.1 dan Gambar 4.2, pada bagian kolom atas adalah a, b, c, e sampai ad dimana merupakan singkatan dari penamaan label jenis bunga. Ditunjukkan pada sisi kanan untuk setiap baris. Secara berurutan adalah bunga bawang sebrang, bunga ceplok piring, bunga delapan dewa sampai dengan yang terakhir adalah bunga kamboja dan bunga kertas. Terlihat bagian diagonal adalah bagian dengan angka-angka yang besar. Menunjukkan secara visual model dapat mengklasifikasi lebih banyak benar daripada yang salah. Bagian yang salah adalah bagian angka yang bukan terletak pada diagonal. Berbeda dengan Gambar 4.1 dan Gambar 4.2, pada Gambar 4.3 mengenali jenis bunga delapan dewa sebanyak 8 berbeda dengan pada Gambar 4.2 yang mengenali jenis bunga delapan dewa atau bunga dengan urutan ke tiga label c yaitu sebanyak 9 jenis yang dapat dikenali dengan benar. Untuk seterusnya dapat dilihat pada Gambar 4.3 dibawah ini.

Gambar *confusion matrix* yang terakhir adalah Gambar 4.4, sebagaimana dijelaskan pada keterangan sebelumnya untuk Gambar 4.1, Gambar 4.2 dan Gambar 4.3, pada bagian kolom atas adalah a, b, c, e sampai ad dimana merupakan singkatan dari penamaan label jenis bunga. Ditunjukkan pada sisi kanan untuk setiap baris. Secara berurutan adalah bunga bawang sebrang, bunga ceplok piring,

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	<-- classified as
31	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	a = bawang sebrang
5	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	b = ceplok piring
0	0	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	c = delapan dewa
0	0	3	4	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = chamaecrista fasciculata
1	0	0	2	17	3	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	e = bunga sepatu putih
0	0	0	0	0	9	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	f = bunga batavia
0	0	0	0	0	5	17	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	g = bugenvil merah muda
0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	h = bunga kamboja
0	0	0	0	0	0	0	1	2	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	i = bugenvil putih
0	0	0	0	1	0	1	0	4	2	0	0	0	0	0	0	1	0	0	1	0	2	0	0	0	0	0	0	0	0	j = bunga kupu-kupu
0	0	0	0	1	0	0	0	0	2	15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	k = Bunga sepatu merah
0	0	0	0	0	0	0	0	0	0	2	17	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	l = Rolia pink
0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = Bunga rumput
0	0	1	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	n = melati
0	0	0	0	1	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = soka kuning
0	0	0	0	0	0	0	0	0	0	1	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	p = soka merah
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	20	0	1	0	0	0	0	0	0	0	0	0	0	1	0	q = dadap merah
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	r = bunga mentega
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	2	0	0	0	0	0	0	0	0	0	0	s = bunga sepatu kuning
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	18	0	1	0	0	0	0	0	0	0	0	0	0	t = bunga merak
0	0	0	0	0																										

Terlihat bagian diagonal adalah bagian dengan angka-angka yang besar. Jelaskan secara visual model dapat mengklasifikasi lebih banyak benar dan yang salah. Bagian yang salah adalah bagian angka yang bukan terletak diagonal. Berbeda dengan Gambar 4.1, Gambar 4.2 dan Gambar 4.3 secara visual dapat mengenali dengan benar jenis bunga bawang sebrang, bunga ceplok dan bunga delapan dewa sebanyak 29, 9 dan 6. Untuk jenis lainnya dari *confusion matrix* dapat dilihat pada Gambar 4.4.

Metode	Evaluasi			
	Akurasi	Presisi	Recall	F-Measure
Warna+Tekstur+BoVW	81.67 %	80.90 %	81.70 %	81.00 %
Warna+Tekstur+WSA	85.50 %	85.30 %	85.50 %	85.00 %
Warna+Tekstur+DVSA	82.33 %	81.20 %	82.30 %	81.60 %
Warna+Tekstur	82.00 %	81.10 %	82.00 %	81.20 %

59

Sedangkan kolom selanjutnya adalah evaluasi, dengan namanya masing-masing adalah akurasi, presisi, recall dan f-measures. Dengan metode dasar adalah metode Warna dan Tekstur, sedangkan metode lainnya adalah metode *word*. Yaitu penggabungan masing-masing dengan *word* BoVW, WSA dan DVSA. Detail hasilnya dapat dilihat pada Tabel 4.3

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad				
29	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0	0	<-- classified as			
S	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	a = bawang sebrang		
	0	6	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	b = ceplok piring		
	0	0	4	2	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	c = delapan dewa		
	0	0	0	1	20	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	d = chamaecrista fasciculata		
	0	0	0	0	1	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	e = bunga sepatu putih		
	0	0	0	0	0	5	15	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	g = bugenvil merah muda		
	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	h = bunga kamboja		
	0	0	0	0	0	0	0	1	2	1	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	i = bugenvil putih		
	0	0	0	0	1	0	0	0	0	6	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	j = bunga kupu-kupu	
	0	0	0	0	1	0	0	0	0	1	15	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	k = Bunga sepatu merah	
	0	0	0	0	0	0	0	0	0	0	2	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	l = Rolia pink	
	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = Bunga rumput	
	0	0	1	0	1	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	n = melati	
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = soka kuning	
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	22	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	p = soka merah	
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	q = dadap merah
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	r = bunga mentega	
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	22	0	2	0	0	0	0	0	0	0	0	0	0	s = bunga sepatu kuning	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	18	0	0	1	0	0	0	0	0	0	0	1	0	t = bunga merak
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	u = widelia biflora	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	1	0	0	0	v = bunga iris kuning	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	w = lili	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	x = bunga tapak dara	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	y = bunga putri malu	
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	z = pseuderanthemum reticulatum	
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	2	0	0	aa = bunga tomat		
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	13	0	0	0	ab = ceplok piring8		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	ac = bunga kamboja jepang		
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	19	0	ad = bunga kertas	

Gambar 4.4 Hasil confusion matrix dengan tambahan fitur word dari BoVW

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Beberapa kesimpulan diberikan sebagai berikut

1. Metode *word* yang diusulkan kompetitif dalam mengenali jenis gambar. Hal ini ditunjukkan dengan hasil performa yang selalu unggul terhadap metode *word* dasar dan lebih unggul untuk jumlah *word* empat dan enam terhadap metode WSA. Metode *word* WSA selalu unggul pada jumlah *word* dua baik dengan metode *word* dasar maupun dengan metode yang diajukan. Hal ini ditemui pada semua skenario uji coba yaitu pada pengenalan gambar umum maupun pada pengenalan gambar bunga. Baik pada pengujian dengan hanya menggunakan *word* maupun dengan penggabungan *word* dengan ekstraksi fitur lain.
2. Metode *word* yang *spatial*, jika digabungkan dengan metode ekstraksi fitur lain seperti ekstraksi fitur warna dan tekstur dapat meningkatkan performa dalam mengenali jenis objek dibandingkan dengan hanya menggunakan ekstraksi fitur saja. Ditunjukkan pada skenario pengenalan jenis objek bunga bahwa metode *spatial* dengan hanya menggunakan sedikit *word* selalu unggul terhadap metode yang menggunakan ekstraksi fitur saja. Sedangkan untuk metode *word* yang menjumlahkan saja, jika digabungkan dengan metode ekstraksi fitur lain dapat menurunkan performa dibandingkan dengan hanya menggunakan ekstraksi fitur saja.

5.2 Saran

Beberapa saran atau *future work* dijelaskan sebagai berikut

1. Saran yang pertama adalah melakukan perhitungan informasi *spatial* lain seperti membentuk kuadran yang tidak dibentuk dari garis horisontal dan garis vertikal. Semisal garis horisontal dan garis vertikal tersebut diputar beberapa derajat. Atau menambahkan kuadran lain seperti daerah yang

dibentuk oleh *spatial pyramid*. Dari kuadran baru inilah informasi *spatial* didapatkan.

2. Saran lain yaitu perhitungan deskriptor bisa lebih dinamis semisal dengan ekstraksi fitur yang populer. Seperti ekstraksi fitur tekstur *local binary pattern* dan ekstraksi fitur warna. Dari ekstraksi fitur yang populer ini kemudian dibentuk deskriptor lokal yang digunakan untuk menghasilkan *word*.
3. Saran selanjutnya adalah melakukan penggabungan hasil informasi *spatial* dan hasil penjumlahan dari *word*. Jadi tidak hanya menggunakan informasi *spatial* saja atau informasi penjumlahan dari *word* saja. Penggabungan inilah yang merupakan fitur vektor akhir dimana masing-masing dari hasil informasi *spatial* dan hasil penjumlahan dari *word* didapatkan dari proses *pooling*.
4. Pada penggabungan pada saran ke tiga proses *pooling* dapat dilakukan dengan *average pooling* maupun *max pooling* untuk metode *word* yang menjumlahkan dari *word* maupun untuk metode yang *spatial*.
5. Pada uji yang terakhir yaitu menguji ketika fitur yang dihasilkan dari metode *word* digabungkan langsung dengan ekstraksi fitur lain. Didapat hasil peningkatan performa untuk penggabungan dengan metode *spatial*. Ke depan dapat dilakukan pembobotan pada hasil dari metode *word* maupun metode ekstraksi fitur umum. Semisal dengan memberikan bobot yang lebih pada fitur yang dihasilkan oleh metode *word*.

DAFTAR PUSTAKA

- Avila, S., Thome, N., Cord, M., Valle, E., de A. Araújo, A., 2013. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding* 117, 453–465.
- A. Vedaldi and B.Fulkerson, 2008. (VLFeat): An Open and Portable Library of Computer Vision Algorithms
- Bolvinou, A., Pratikakis, I., Perantonis, S., 2013. Bag of spatio-visual words for context inference in scene classification. *Pattern Recognition* 46, 1039–1053.
- Dimitrovski, I., Kocev, D., Loskovska, S., Džeroski, S., 2014. Fast and efficient visual codebook construction for multi-label annotation using predictive clustering trees. *Pattern Recognition Letters* 38, 38–45.
- Koniusz, P., Yan, F., Mikolajczyk, K., 2013. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding* 117, 479–492.
- Li, Z., Yap, K.-H., 2013. An efficient approach for scene categorization based on discriminative codebook learning in bag-of-words framework. *Image and Vision Computing* 31, 748–755.
- López-Sastre, R.J., García-Fuertes, A., Redondo-Cabrera, C., Acevedo-Rodríguez, F.J., Maldonado-Bascón, S., 2013. Evaluating 3D spatial pyramids for classifying 3D shapes. *Computers & Graphics* 37, 473–483.
- Penatti, O.A.B., Silva, F.B., Valle, E., Gouet-Brunet, V., Torres, R. da S., 2014. Visual *word* spatial arrangement for image retrieval and classification. *Pattern Recognition* 47, 705–720.
- Sánchez, J., Perronnin, F., de Campos, T., 2012. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters* 33, 2216–2223.
- Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A., 2013. Fisher vector faces in the wild, in: *British Machine Vision Conference*. p. 7.
- Wang, J., Liu, P., She, M.F.H., Nahavandi, S., Kouzani, A., 2013. Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control* 8, 634–644.

- Zagoris, K., Pratikakis, I., Antonacopoulos, A., Gatos, B., Papamarkos, N., 2014. Distinction between handwritten and machine-printed text based on the bag of visual *word* model. *Pattern Recognition* 47, 1051–1062.
- Zhang, C., Wang, S., Huang, Q., Liu, J., Liang, C., Tian, Q., 2013. Image classification using spatial pyramid robust sparse coding. *Pattern Recognition Letters* 34, 1046–1052.

BIODATA PENULIS



Gama Wisnu Fajarianto, lahir di Jember pada tanggal 13 Nopember 1988. Penulis telah menyelesaikan studi S1 sebagai Sarjana Komputer (S.Kom.) di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Pada tahun 2012 penulis mendapat kesempatan untuk melanjutkan studi S2 di Program Sarjana Teknik Informatika ITS dengan beasiswa Freshgraduate. Pada Januari 2015 penulis telah mengikuti ujian Tesis sebagai syarat mendapatkan gelar Magister Komputer di Institut Teknologi Sepuluh Nopember Surabaya. Penulis mengambil bidang keahlian Komputasi Cerdas dan Visualisasi dan dapat dihubungkan melalui email gamawifa@gmail.com